



院 系 数据科学与计算机学院 学 号 15352237 姓 名 刘顺宇

【实验题目】JS 实验 2

【实验目的】学习 Javascript 高级编程。

【调试方法】

方法 1: chrome 调试 js 的方法: [参考](#)

方法 2:

```
try{  
    要调试的代码  
}  
catch(e){  
    alert(e);  
}
```

【实验环境】

- (1) 自己安装的 tomcat
- (2) 自己安装的 Eclipse
- (3) 在线调试: <http://172.18.187.233:8080/testjsp/>
要用的图像在 <http://172.18.187.233:8080/testjsp/temp/images> 中

【实验内容】

1、设计两个数字计时器(timer1.html, timer2.html)。

(1) 简单计时器: 按启动键开始一直计时。

提示: `setTimeout(timer, 1000);` //ms

参考资料: JavaScript.ppt (下) 定时器

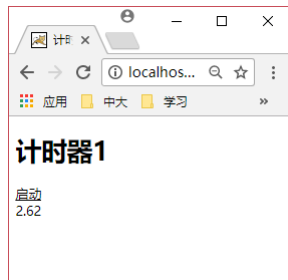
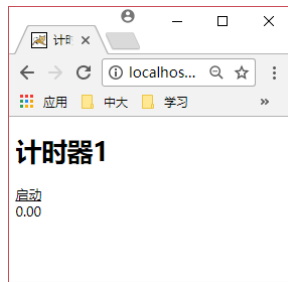
参考截屏:



视频演示见 timer1.mp4:



完成后截屏：



源码(timer1.html):

```
1 <!DOCTYPE html>
2 <html lang="zh-cn">
3 <head>
4 <title>计时器1</title>
5 <style>
6   span {text-decoration:underline;}
7   span:hover {cursor:pointer}
8 </style>
9 <script>
10   var t = 0.00;
11   function time(){
12     t = t + 0.01;
13     var timer = document.getElementById("timer");
14     timer.innerHTML = t.toFixed(2);
15     window.setTimeout(time, 10);
16   }
17 </script>
18 </head>
19 <body>
20 <h1>计时器1</h1>
21 <div><span id="start" onclick="time()">启动</span></div>
22 <div id="timer">0.00</div>
23 </body>
24 </html>
```

(2) 复杂计时器：按启动开始计时，“启动变”为“暂停”，再按继续启动。再暂停时按复位按钮将回到初始状态。

参考截屏：

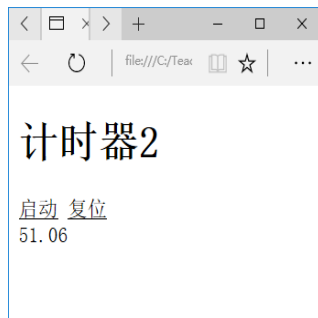




点击启动后：



点击暂停后

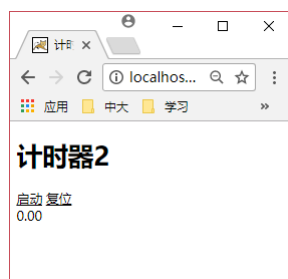


点击复位后

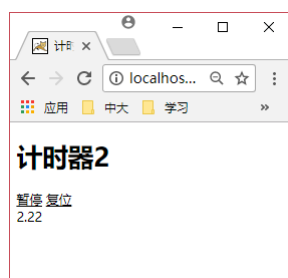


视频演示见 timer2.mp4:

完成后截屏：

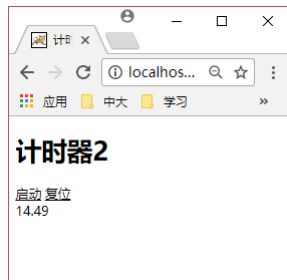


点击启动后：

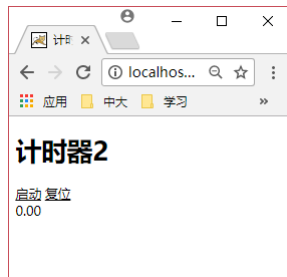




点击暂停后：



点击复位后：



源码(timer2.html):

```
1 <!DOCTYPE html>
2 <html lang="zh-cn">
3 <head>
4   <title>计时器2</title>
5   <style>
6     span {text-decoration:underline;}
7     span:hover {cursor:pointer}
8   </style>
9   <script>
10    var t = 0.00;
11    var start = false;
12    function toggleTimer(){
13      if(!start){
14        start = true;
15        var button = document.getElementById("start");
16        button.innerHTML = "暂停";
17        time();
18      }else{
19        start = false;
20        var button = document.getElementById("start");
21        button.innerHTML = "启动";
22        time();
23      }
24    }
25    function time(){
26      if(start){
27        t = t + 0.01;
28        var timer = document.getElementById("timer");
29        timer.innerHTML = t.toFixed(2);
30        window.setTimeout(time, 10);
31      }
32    }
33
34    function resetTimer(){
35      start = false;
36      t = 0.00;
37      var timer = document.getElementById("timer");
38      timer.innerHTML = t.toFixed(2);
39      var button = document.getElementById("start");
40      button.innerHTML = "启动";
41    }
42  </script>
43 </head>
44 <body>
45   <h1>计时器2</h1>
46   <div>
47     <span id="start" onclick="toggleTimer()">启动</span>
48     <span onclick="resetTimer()">复位</span></div>
49   <div id="timer">0.00</div>
50 </body>
</html>
```



2、(validate.html)验证密码，输入两个密码，如果不相同，则弹出出错信息，否则，提交到 OK.jsp (没有该文件时会出错)。

参考资料：JavaScript.ppt（下）事件对象-通过事件提交表单

提示：onclick="return submit1();" submit1()返回 false 会取消默认行为。

参考截屏：

输入 123 和 1234:

按提交按钮弹出信息，并取消了提交：

如果输入相同，则会提交，提交可能会出错（OK.jsp 不存在）。



完成后截屏：



实验报告

输入 123 和 1234:

按提交按钮弹出信息，并取消了提交：

A screenshot of a web browser window. The address bar shows 'localhost:8080'. The page content displays the text 'localhost:8080 显示:' followed by '两次输入的密码不相等' (The two entered passwords do not match). Below this, there is a text input field containing '重输入密码: ****' (Re-enter password: ****) and a button labeled '提交' (Submit).

如果输入相同，则会提交，提交可能会出错（OK.jsp 不存在）。

The screenshot shows a web browser window with the address bar displaying 'local...'. The main content area shows an 'HTTP Status 404 - /js2/OK.jsp' error. Below the title, there is a 'Type' field with the value 'Status report', a 'message' field with the value '/js2/OK.jsp', and a 'description' field with the text 'The requested resource is not available.' The browser's user agent string at the bottom is 'Apache Tomcat/8.0.47'.

源码(validate.html):

[illegible]



3、制作幻灯片显示（参考 html 实验 8 的第 4 题）



功能说明：

- (1) 用 `setTimeout` 制作幻灯片动画。装载网页后自动播放幻灯片。具体见 `slider.mp4`
- (2) 当 `mouseover` 图片时，显示左右箭头，并停止动画。当 `mouseout` 画面时，隐藏左右箭头，重新启动动画。
- (3) 每次点击左箭头将移动一副图像。
- (4) 右下角的三个点每个点对应一幅图像，当前图像的变化时所对应的点也会变化。点击它们时会显示对应的图像。

参考资料：Javascript(下)v2.pdf 事件

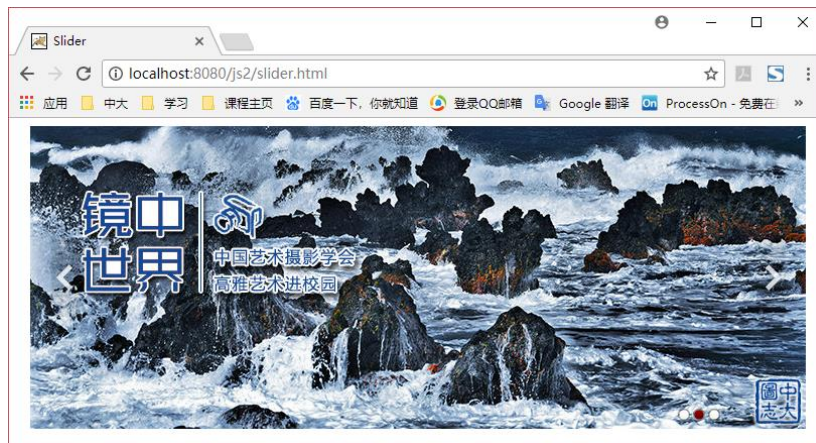
参考视频：slider.mp4

编程参考：

- (1) 图片采用三个 `img`，并用一个 `div` 包含，样式 `white-space` 要加上 `nowrap`。
 - (2) 外层再用一个 `div`，设置宽度为一个图片的宽度，并设置 `overflow` 为 `hidden`。
 - (3) 上面的两个 `div` 均为 `relative`。
 - (4) 每个点和箭头采用 `div` 和背景图实现，三个点再采用一个 `div` 封装，该 `div` 和两个箭头的 `div` 均采用绝对定位。
 - (5) 通过增加 `class` 或删除 `class` 实现三个点中活动点的切换。
- * 所用方法: `querySelector()`, `querySelectorAll()`, `classList.add()`, `classList.remove()`

完成后的屏幕截图：





完成后的 html 源码(slider.html 或 slider.jsp):

```
1 <!doctype html>
2 <html>
3 <head>
4 <title>Slider</title>
5 <style>
6     #slideMain {
7         width:770px;
8         height: 300px;
9         margin: 0 auto;
10        position:relative;
11        overflow: hidden;
12    }
13    .slides {
14        overflow: hidden;
15        position: absolute;
16        width:3080px;
17    }
18    .slide {
19        overflow: hidden;
20        width:770px;
21        height: 300px;
22        margin:0px;
23        padding:0px;
24        display: block;
25        float: left;
26    }
27 }
28
29 #arrow1{
30     position: absolute;
31     top:110px;
32     left:20px;
33     height:80px;
34     width:30px;
35     overflow:hidden;
36 }
```




```
37 #arrow2{
38     position: absolute;
39     top:110px;
40     right:20px;
41     height:80px;
42     width:30px;
43     overflow:hidden;
44 }
45 #arrowimg2{
46     position: absolute;
47     top:-80px;
48 }
49
50 #point1{
51     position: absolute;
52     top:280px;
53     right:115px;
54     height:12px;
55     width:12px;
56     overflow:hidden;
57 }
58 #pointimg1{
59     position: absolute;
60     top:-24px;
61 }
62 #point2{
63     position: absolute;
64     top:280px;
65     right:100px;
66     height:12px;
67     width:12px;
68     overflow:hidden;
69 }
70 #pointimg2{
71     position: absolute;
72     top:-0px;
73 }
```

```
74 #point3{
75     position: absolute;
76     top:280px;
77     right:85px;
78     height:12px;
79     width:12px;
80     overflow:hidden;
81 }
82 #pointimg3{
83     position: absolute;
84     top:-0px;
85 }
86
87 .arrow:hover,.point:hover{
88     cursor:pointer;
89 }
90 </style>
91 </head>
```

```
92 <body>
93     <div id="slideMain">
94         <div class="slides" id="slides">
95             
96             
97             
98             
99         </div>
100     <div id="arrow1">
101         
102     </div>
103     <div id="arrow2">
104         
105     </div>
106     <div id="point1">
107         
108     </div>
109     <div id="point2">
110         
111     </div>
112     <div id="point3">
113         
114     </div>
115 </div>
116 </body>
```



```
117 <script type="text/javascript">
118   var slides = document.getElementById("slides");
119   var slideMain = document.getElementById("slideMain");
120   var a1 = document.getElementById("arrowimg1");
121   var a2 = document.getElementById("arrowimg2");
122   var p1 = document.getElementById("pointimg1");
123   var p2 = document.getElementById("pointimg2");
124   var p3 = document.getElementById("pointimg3");
125   var stop = 0;
126   var currentimg = 0;
127   var hover = false;
128
129   function show(){
130     a1.style.visibility="visible";
131     a2.style.visibility="visible";
132     p1.style.visibility="visible";
133     p2.style.visibility="visible";
134     p3.style.visibility="visible";
135     hover = true;
136   }
137
138   function hide(){
139     a1.style.visibility="hidden";
140     a2.style.visibility="hidden";
141     p1.style.visibility="hidden";
142     p2.style.visibility="hidden";
143     p3.style.visibility="hidden";
144     hover = false;
145   }
146
147   slideMain.addEventListener('mouseover', show, false);
148   slideMain.addEventListener('mouseout', hide, false);
149   hide();
150
```

```
151   function slide(){
152     if(hover == false){
153       if(stop > 0){
154         stop = stop - 1;
155       }else{
156         if((currentimg == 0) || (currentimg == -770) || (currentimg == -1540)){
157           stop = 100;
158           currentpoint(1-currentimg/770);
159         }else if(currentimg == -2310){
160           currentimg = 0;
161           stop = 100;
162           currentpoint(1);
163         }
164         slides.style.left = currentimg + "px";
165         currentimg = currentimg - 5;
166       }
167     }
168     setTimeout(slide, 10);
169   }
170   slide();
171
```

```
172   function currentpoint(pos){
173     if(pos == 1){
174       p1.style.top = "-24px";
175       p2.style.top = "-0px";
176       p3.style.top = "-0px";
177       currentimg = 0;
178       slides.style.left = currentimg + "px";
179     }else if(pos == 2){
180       p1.style.top = "-0px";
181       p2.style.top = "-24px";
182       p3.style.top = "-0px";
183       currentimg = -770;
184       slides.style.left = currentimg + "px";
185     }else if(pos == 3){
186       p1.style.top = "-0px";
187       p2.style.top = "-0px";
188       p3.style.top = "-24px";
189       currentimg = -1540;
190       slides.style.left = currentimg + "px";
191     }
192   }
193
```

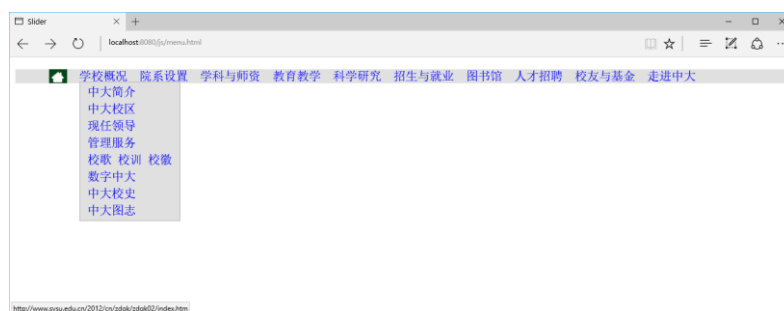


```
172 function currentpoint(pos){
173     if(pos == 1){
174         p1.style.top = "-24px";
175         p2.style.top = "-0px";
176         p3.style.top = "-0px";
177         currentimg = 0;
178         slides.style.left = currentimg + "px";
179     }else if(pos == 2){
180         p1.style.top = "-0px";
181         p2.style.top = "-24px";
182         p3.style.top = "-0px";
183         currentimg = -770;
184         slides.style.left = currentimg + "px";
185     }else if(pos == 3){
186         p1.style.top = "-0px";
187         p2.style.top = "-0px";
188         p3.style.top = "-24px";
189         currentimg = -1540;
190         slides.style.left = currentimg + "px";
191     }
192 }
193 }
```

```
194 function clickarrow(pos){
195     if((currentimg <= 0 && currentimg > -770) || (currentimg <= -2310)){
196         if(pos == 1){
197             currentpoint(3);
198             currentimg = -1540;
199             slides.style.left = currentimg + "px";
200         }else if(pos == 2){
201             currentpoint(2);
202             currentimg = -770;
203             slides.style.left = currentimg + "px";
204         }
205     }else if((currentimg <= -770 && currentimg > -1540)){
206         if(pos == 1){
207             currentpoint(1);
208             currentimg = 0;
209             slides.style.left = currentimg + "px";
210         }else if(pos == 2){
211             currentpoint(3);
212             currentimg = -1540;
213             slides.style.left = currentimg + "px";
214         }
215     }else if((currentimg <= -1540 && currentimg > -2310)){
216         if(pos == 1){
217             currentpoint(2);
218             currentimg = -770;
219             slides.style.left = currentimg + "px";
220         }else if(pos == 2){
221             currentpoint(1);
222             currentimg = 0;
223             slides.style.left = currentimg + "px";
224         }
225     }
226 }
227 }
```

```
228 p1.addEventListener("click", function(){currentpoint(1);}, false);
229 p2.addEventListener("click", function(){currentpoint(2);}, false);
230 p3.addEventListener("click", function(){currentpoint(3);}, false);
231 a1.addEventListener("click", function(){clickarrow(1);}, false);
232 a2.addEventListener("click", function(){clickarrow(2);}, false);
233 </script>
234 </html>
```

4、用 JavaScript 制作菜单（参考 html 实验 8 的第 3 题）：





参考资料: javascript(下)v2.pdf 事件

参考视频: menu.mp4

要求和说明:

在不改动 html 的 body 内的已设计元素的情况下, 如, 不能为元素增加类和 id, 实现下面功能:

- (1) 先用 css 制作好下拉菜单。
- (2) 然后取消 css 的 hover 功能, 并用 mouseover 和 mouseout 进行实现。

编程参考:

- (1) home 图片采用背景图实现
- (2) 通过 document.querySelectorAll("#nav>li")取到主菜单 (数组)。
- (3) 通过闭包和 addEventListener 设置鼠标 mouseover 和 mouseout 事件:
循环执行函数(function(ind){})(i)来绑定每个菜单项的事件

完成后的屏幕截图:



完成后的 html 源码(menu.html 或 menu.jsp):

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>menu</title>
5 <base href="http://www.sysu.edu.cn/2012/cn/">
6 <style>
7     a:link,a:visited {text-decoration:none;color:blue}
8     * { margin:0; padding:0;font-size:18px;font-family:宋体;line-height:1.2em}
9
10    #nav{
11        margin-top: 20px;
12    }
13    li {list-style-type:none;}
14    #nav>li {
15        background-color: rgb(200,200,200);
16        width:9em;
17        height:1.5em;
18        float: left;
19        line-height: 1.5em;
20    }
```



```
21 #nav>li>ul{
22     width:9em;
23     background-color: rgb(200,200,200);
24     border: 1px solid #AAAAAA;
25     display:none;
26 }
27 #nav>li>ul>li{
28     padding-left: 10px;
29     height: 1.8em;
30     line-height: 1.8em;
31 }
32 .home{
33     background: url('http://localhost:8080/js2/images/home.gif') no-repeat;
34     background-repeat: no-repeat;
35     background-size: contain;
36     background-origin: padding-box;
37     padding-top: 3px;
38     padding-bottom: 5px;
39 }
40 </style>
41 <script>
42 </script>
43 </head>
```

```
108 <script>
109     var li=document.querySelectorAll("#nav>li");
110     li[0].style.textAlign = "center";
111     li[0].style.marginLeft = "10px";
112     for(var i=1; i<11; i++){
113         (function(i){
114             li[i].addEventListener("mouseover",function(){li[i].childNodes[2].style.display="
115                 inline-block";},false);
116             li[i].addEventListener("mouseout",function(){li[i].childNodes[2].style.display="
117                 none";},false);
118         })(i);
119     }
120 </script>
121 </body>
122 </html>
```

(body 部分代码没有做任何修改由于太长不再截图)

5、DOM 用正文为空的 dom.html，通过 javascript 的 DOM 操作，实现以下 html 网页(jys.html):



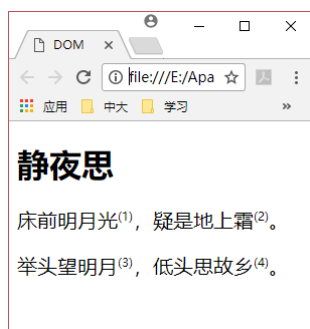
参考资料: javascript(下).ppt 文档对象模型

实验要求:

- 1、所有节点的创建采用 document.createElement()、document.createTextNode()和 e.cloneNode()实现；加入元素采用 e.appendChild()实现。* 不可以使用 innerHTML。
可以用 document.body.appendChild(e);把元素加入当前网页。
- 2、先设计第一行，除了第一个上标，第二上标采用克隆第一个上标实现。
- 3、上标中的属性采用 e.style.textAlign="super"和 e.style.fontSize="12px"实现
- 4、第二行采用克隆第一行实现，利用.childNodes 和 nodeValue 修改文本节点的内容。
* 其中的 e 表示某个元素节点。



完成后的屏幕截图：



完成后的源码(dom.html 或 dom.jsp):

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>DOM</title>
5 <style>
6 p {font-size:20px}
7 </style>
8 </head>
9 <body>
10 </body>
11 <script type="text/javascript">
12     var h1 = document.createElement("h1");
13     var title = document.createTextNode("静夜思");
14     h1.appendChild(title);
15     document.body.appendChild(h1);
16
17     var p1 = document.createElement("p");
18     var text1 = document.createTextNode("床前明月光");
19     var span1 = document.createElement("span");
20     var text2 = document.createTextNode("(1)");
21     span1.appendChild(text2);
22     span1.style.verticalAlign = "super";
23     span1.style.fontSize = "12px";
24     var text3 = document.createTextNode(", 疑是地上霜");
25     var span2 = span1.cloneNode(true);
26     span2.childNodes[0].nodeValue="(2)";
27     var text4 = document.createTextNode("。");
28     p1.appendChild(text1);
29     p1.appendChild(span1);
30     p1.appendChild(text3);
31     p1.appendChild(span2);
32     p1.appendChild(text4);
33
34     var p2 = p1.cloneNode(true);
35     p2.childNodes[0].nodeValue="举头望明月";
36     p2.childNodes[1].childNodes[0].nodeValue="(3)";
37     p2.childNodes[2].nodeValue="， 低头思故乡";
38     p2.childNodes[3].childNodes[0].nodeValue="(4)";
39     p2.childNodes[4].nodeValue="。";
40
41     document.body.appendChild(p1);
42     document.body.appendChild(p2);
43 </script>
44 </html>
```

6、用 ajax-get 技术实现查询功能。

参考资料：javascript(下)v2.pdf AJAX 技术-GET

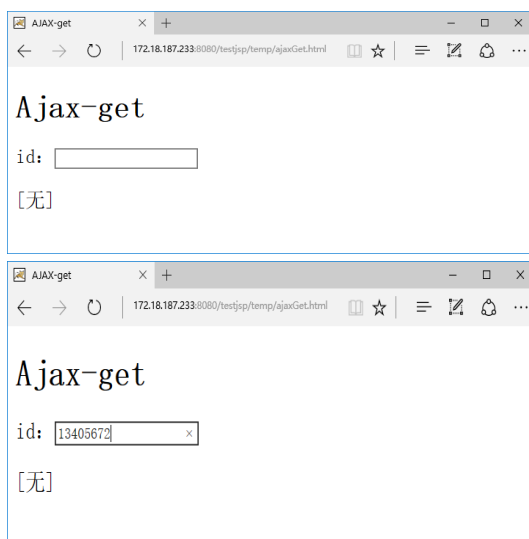
参考例程：JSP 实验(2)

实验要求：

用 ajax-get 技术(ajaxGet.html 和 get.jsp)实现以下功能：



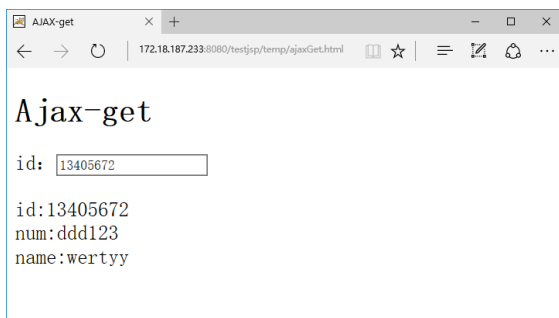
(1)在下面画面中输入 id:



(2) 离开 id 输入域后利用 ajax-get 技术调用 get.jsp 从 stu 表中取得 num 和 name 并显示出来，即 alert(xmlhttp.responseText)，返回的格式见下图：



(3) 关闭弹出框后:

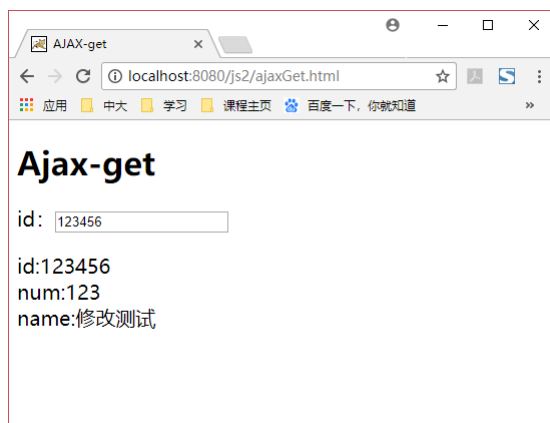
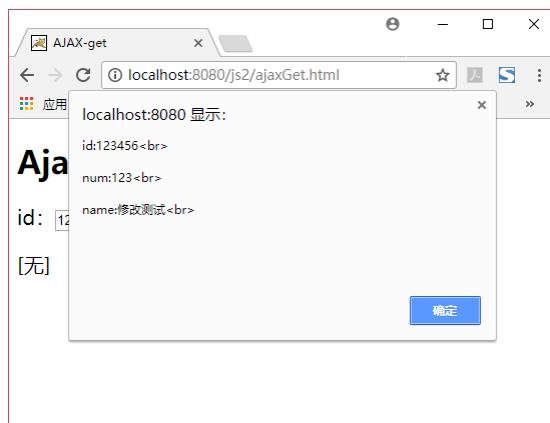
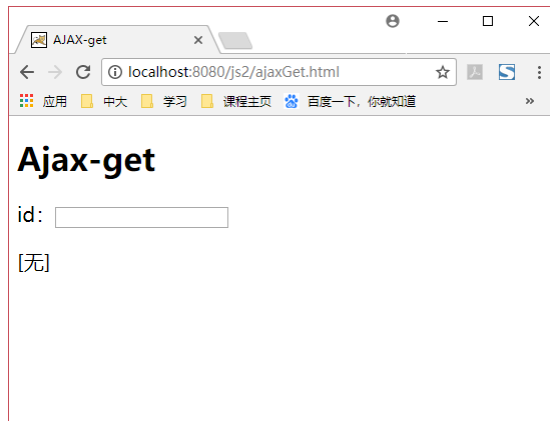


实验说明:

- (1) 用 <http://172.18.187.233:8080/jsp/browseStu.jsp> 查询数据。
MySQL 服务器: 172.18.187.233:53306 用户名:user 密码:123
数据库: teaching 表: stu 字段:id,num,name
- (2) 编写 get.jsp 程序可参考 JSP 实验(2)的 browseStu2.jsp, 在线调试时可以在文件名加上学号 get15000001.jsp, 以免冲突。
- (3) get.jsp 修改后有时要在浏览器中直接运行并刷新才生效。
例如: <http://172.18.187.233:8080/testjsp/temp/get.jsp>
- (4) 离开输入域的事件为 blur, 可以采用属性 onblur 触发。



完成后截屏：



源码 ajaxGet.html 或 ajaxGet.jsp:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>AJAX-get</title>
5   <style>
6     p {font-size:20px}
7   </style>
8 </head>
9 <body>
10  <h1>Ajax-get</h1>
11  <p>id: <input id="t1" type="text" onblur="get()"></p>
12  <p id="t2">[无]</p>
```




```
13 <script>
14     function get() {
15         var xmlhttp = new XMLHttpRequest();
16         xmlhttp.onreadystatechange = function () {
17             if (xmlhttp.readyState == 4) {
18                 if (xmlhttp.status >= 200 && xmlhttp.status < 300 || xmlhttp.status == 304) {
19                     alert(xmlhttp.responseText);
20                     var oTest = document.getElementById("t2");
21                     oTest.innerHTML = xmlhttp.responseText;
22                 } else {
23                     alert("error");
24                 }
25             };
26         };
27         xmlhttp.open("get", "get.jsp?id=" + document.getElementById("t1").value, true);
28         xmlhttp.send(null);
29     }
30 </script>
31 </body>
32 </html>
```

源码 get.jsp:

```
1 <%@ page language="java" import="java.util.*,java.sql.*"
2     contentType="text/html; charset=utf-8"
3 %><%
4     request.setCharacterEncoding("utf-8");
5     String connectString = "jdbc:mysql://172.18.187.233:53306/teaching" + "
6     ?autoReconnect=true&useUnicode=true&characterEncoding=UTF-8&useSSL=false";
7     String table=new StringBuilder("");
8     try{
9         Class.forName("com.mysql.jdbc.Driver");
10        Connection con = DriverManager.getConnection(connectString, "user", "123");
11        Statement stmt = con.createStatement();
12        ResultSet rs = stmt.executeQuery(String.format("select * from stu where id = '%s'",
13            request.getParameter("id")));
14        if(rs.next()){
15            out.println("id:" + rs.getString("id") + "<br>");
16            out.println("num:" + rs.getString("num") + "<br>");
17            out.println("name:" + rs.getString("name") + "<br>");
18        }else{
19            out.println("[无]");
20        }
21        rs.close();
22        stmt.close();
23        con.close();
24    }
25    catch (Exception e){
26        System.out.println(e.getMessage());
27    }
28 %>
```

7、用 ajax-post 技术(xmlhttpPost.html, post.jsp)实现修改数据的功能。

参考资料: Javascript(下)v2.pdf AJAX 技术-POST

实验要求:

(1) 先浏览记录:

id	学号	姓名	-
13405651	174391	16791	修改 删除
13405656	111112	56884	修改 删除
13405672	ddd123	aaaaa	修改 删除
13405673	1236ss	qqqqq	修改 删除

[新增](#) [上一页](#) [下一页](#)



(2) 选择输入某个 id 后:

Ajax-post

id:

姓名:

(3) 当输入新的姓名并离开(onblur)时用 ajax 技术调用 post.jsp 修改具有该 id 记录的姓名的值。

Ajax-post

id:

姓名:

修改成功!

(4) 查看修改后的情况:

浏览学生名单

id	学号	姓名	-
13405651	174391	16791	修改 删除
13405656	111112	56884	修改 删除
13405672	ddd123	张某	修改 删除
13405673	1236ss	qqqqq	修改 删除

[新增](#) [上一页](#) [下一页](#)

实验说明:

- (1) 编写 post.jsp 程序可参考 JSP 实验 updateStu.jsp, 在线调试时可以在文件名加上学号 post13000001.jsp, 以避免重名冲突。
- (2) 在线调试用 <http://172.18.187.233:8080/jsp/browseStu.jsp> 查看记录数据和修改结果

完成后截屏:

Ajax-post

id:

姓名:



AJAX-post

id:

姓名:

修改成功!

修改前后数据库对比:

信息	结果1	概况	状态
id	num	name	
123456	123	修改测试	
13405703	111	陈某某	
15351234	123454321	newjs2	
15352029	15352029	把把	
111111125	64654	哈哈哈哈哈	
111111126	879654	qqq	

信息	结果1	概况	状态
id	num	name	
123456	123	123	
13405703	111	陈某某	
15351234	123454321	newjs2	
15352029	15352029	把把	
111111125	64654	哈哈哈哈哈	
111111126	879654	qqq	

源码(`ajaxPost.html` 或 `ajaxPost.jsp`):

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>AJAX-post</title>
5   <style>
6     p {font-size:20px}
7   </style>
8 </head>
9 <body>
10  <h1>Ajax-post</h1>
11  <p>id: <input id="t1" type="text"></p>
12  <p>姓名: <input id="t2" type="text" onblur="post()"></p>
13  <p><span id="t4"></span></p>
14  <script>
15    function post() {
16      var xmlhttp = new XMLHttpRequest();
17      xmlhttp.onreadystatechange = function () {
18        if (xmlhttp.readyState == 4) {
19          if (xmlhttp.status >= 200 && xmlhttp.status < 300 || xmlhttp.status >= 304) {
20            alert(xmlhttp.responseText);
21            var oTest = document.getElementById("t4");
22            oTest.innerHTML = xmlhttp.responseText;
23          } else {
24            alert("error");
25          }
26        }
27      };
28      var t1 = document.getElementById("t1").value;
29      var t2 = document.getElementById("t2").value;
30      var param = "id=" + encodeURIComponent(t1) + "&name=" + encodeURIComponent(t2);
31      xmlhttp.open("post", "post.jsp", true);
32      xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
33      xmlhttp.send(param);
34    }
35  </script>
36 </body>
37 </html>
```



源码(post.jsp):

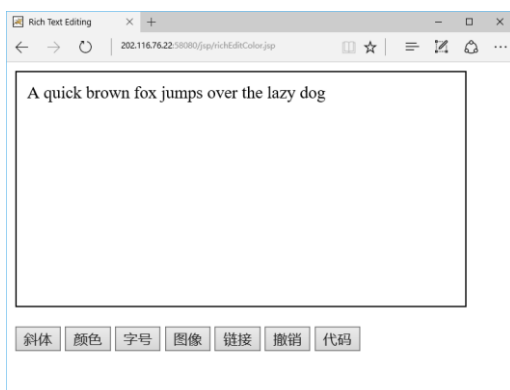
```
1 <%@ page language="java" import="java.util.*,java.sql.*"
2     contentType="text/html; charset=utf-8"
3 %><%request.setCharacterEncoding("utf-8");
4     String connectString = "jdbc:mysql://172.18.187.233:53306/teaching" + "
5     ?autoReconnect=true&useUnicode=true&characterEncoding=UTF-8&useSSL=false";
6     StringBuilder table=new StringBuilder("");
7     try{
8         Class.forName("com.mysql.jdbc.Driver");
9         Connection con = DriverManager.getConnection(connectString, "user", "123");
10        Statement stmt = con.createStatement();
11        int rs = stmt.executeUpdate(String.format("UPDATE stu SET name = '%s' WHERE id = '%s'"
12        , request.getParameter("name"), request.getParameter("id")));
13        if(rs>0) out.println("修改成功!");
14        else out.println("修改失败!");
15        stmt.close();
16        con.close();
17    }
18    catch (Exception e){
19        System.out.println(e.getMessage());
20    }
21 %>
```

8、采用 div 实现以下富文本编辑。

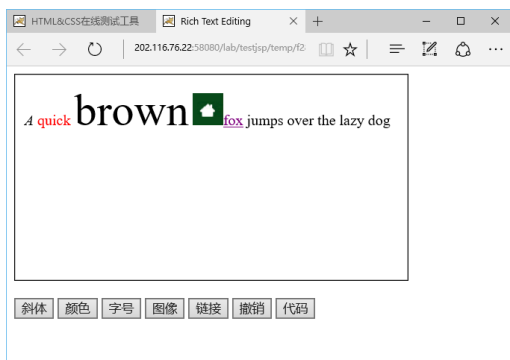
参考资料: Javascript(下)v2.pdf p49-p51 和附录 4

实验要求:

(1) 运行开始:



(2) 依次选中一个词进行设置格式 (图像加当前词在后面): (*字号为 7)

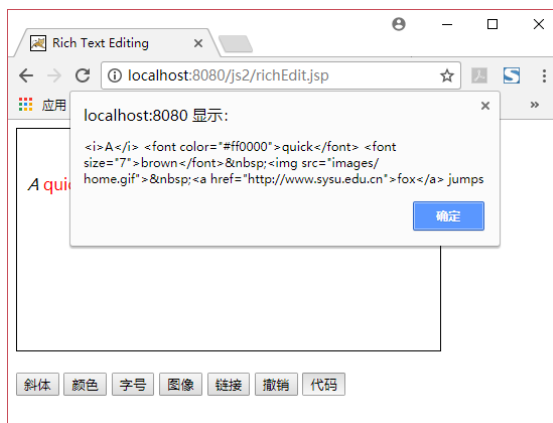
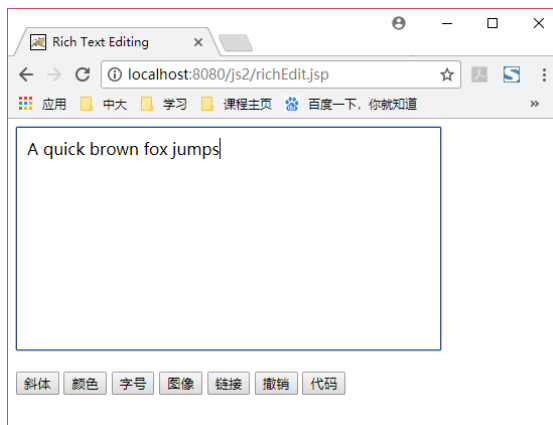




(3) 按了撤销按钮后可以撤销前面的操作，按了代码按钮后会显示编辑框中实际的代码：



完成后截屏：





源码 richEdit.jsp:

```
1 <%@ page language="java" import="java.util.*,java.io.*"
2 contentType="text/html; charset=utf-8"%>
3 <!DOCTYPE html>
4 <html><head><title>Rich Text Editing</title></head>
5 <body>
6   <div class="editable" id="richedit" contenteditable="true" style="padding:10px; width:400
   px; height:200px; border:solid 1px black;">
7   </div><br>
8   <input type="button" value="斜体" onclick="italic()">
9   <input type="button" value="颜色" onclick="forecolor()">
10  <input type="button" value="字号" onclick="fontsize()">
11  <input type="button" value="图像" onclick="insertimage()">
12  <input type="button" value="链接" onclick="createlink()">
13  <input type="button" value="撤销" onclick="undo()">
14  <input type="button" value="代码" onclick="show()">
15  <script>
16    function italic() {
17      document.execCommand("italic", false, null);
18    }
19    function forecolor() {
20      document.execCommand("forecolor", false, "red");
21    }
22    function fontsize() {
23      document.execCommand("fontsize", false, 7);
24    }
25    function insertimage() {
26      document.execCommand("insertimage", false, "images/home.gif");
27    }
28    function createlink() {
29      document.execCommand("createlink", false, "http://www.sysu.edu.cn");
30    }
31    function undo() {
32      document.execCommand("undo", false, null);
33    }
34    function show() {
35      var o = document.getElementById("richedit");
36      alert(o.innerHTML);
37    }
38  </script>
39 </body>
40 </html>
```

9、(message.html)编写一个消息框弹出程序，当无输入值时显示提示信息，否则显示输入值。要求消息框定位在浏览器客户区中央（参见 position.html），并且整个窗口(包括被遮蔽的网页部分)变灰且禁止操作（参见 opacity.html），直到点击信息框的 OK 按钮才变正常。消息框要求有标题栏和关闭按钮。只改样式和脚本部分。

* 采用 querySelector()选择元素。

说明：

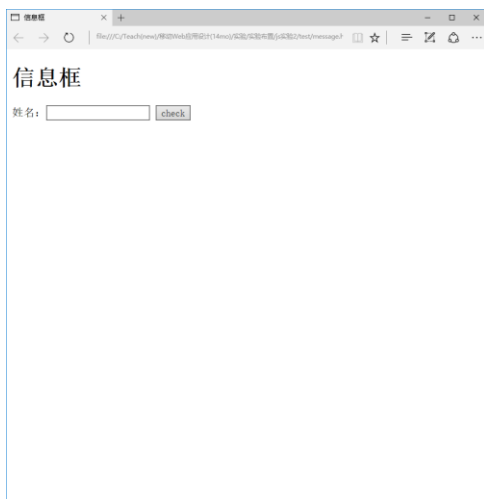
document.documentElement.clientHeight 浏览器客户区的高度

document.documentElement.scrollHeight 整个网页的高度（包含滚出客户区的部分）

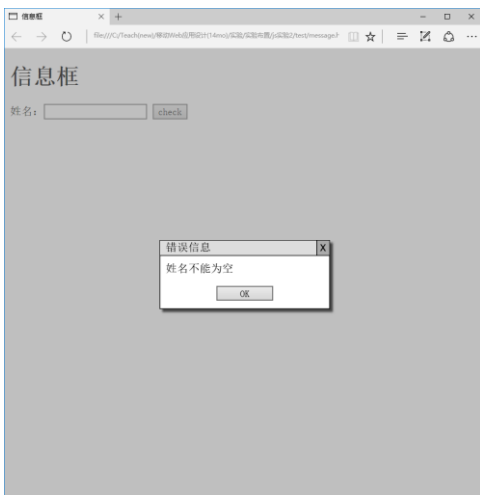




参考显示:



点击 check 之后

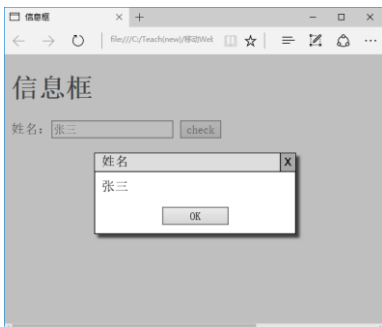


点击 OK, 输入姓名:



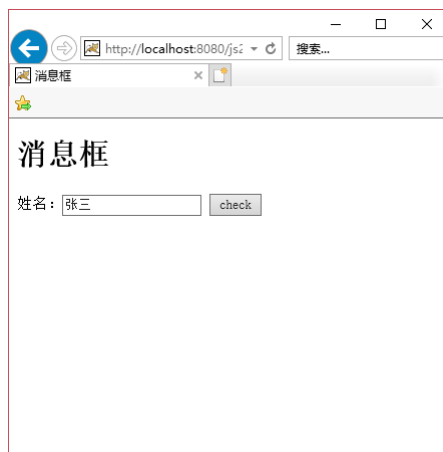
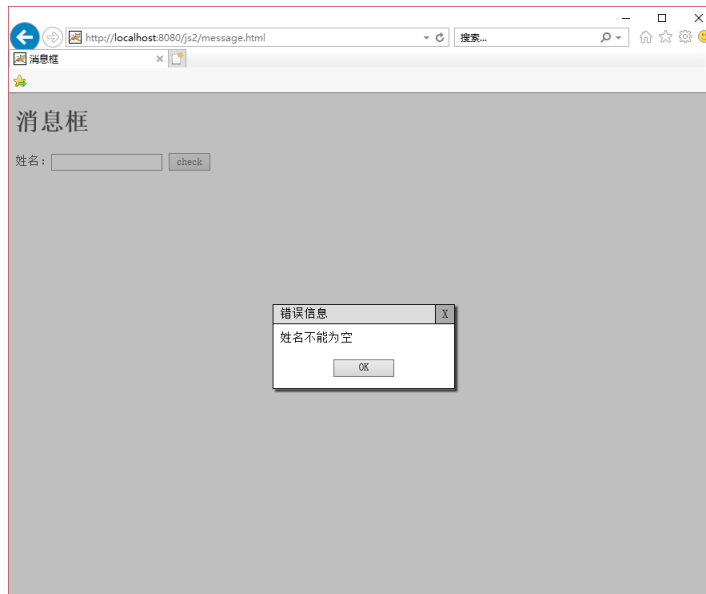
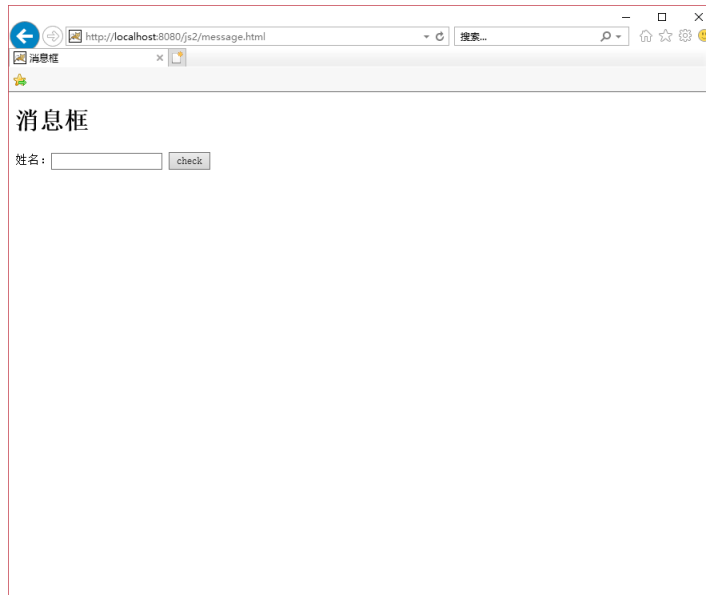
(窗口变小了)

点击 check 后:

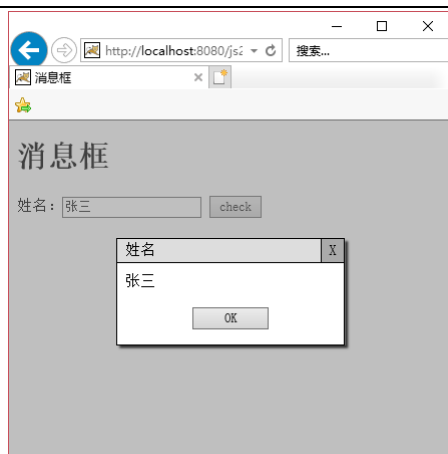




完成实验后截屏：



(点击 OK 之后窗口自动变小)



源程序 (message.html):

```
1 <!DOCTYPE html>
2 <html lang="zh-cn">
3 <head>
4 <title>消息框</title>
5 <style>
6 * {font-family:"宋体"}
7 #shadow {
8 position:absolute;
9 top:0;
10 left:0;
11 z-index:80;
12 background-color:gray;
13 opacity:0.5;
14 display:none;}
15
16 #msgbox1{
17 z-index:800;
18 display:none;
19 position:fixed;
20 background:#FFFFFF;
21 box-sizing:border-box;
22 border: 1px solid black;
23 box-shadow:3px 3px 3px #333333;
24 width:15em;
25 height:7em;
26 }
27
28 .msgheader{
29 height:1.5em;
30 border-bottom: 1px solid black;
31 padding-left: 0.5em;
32 background:rgb(221,221,221);
33 }
34 .msgtitle{
35 float: left;
36 line-height: 1.5em;
37 }
38 .msgclose{
39 float: right;
40 border-left: 1px solid black;
41 line-height: 1.5em;
42 width: 1.5em;
43 text-align: center;
44 background:rgb(170,170,170);
45 }
46 .msgcontent{
47 margin-left: 0.5em;
48 margin-top: 10px;
49 }
50 .btns{
51 margin-top: 18px;
52 text-align: center;
53 }
54 .ok{
55 width: 6em;
56 }
57 </style>
58 </head>
```



```
59 <body>
60 <h1>消息框</h1>
61 <script>
62     function showMsgBox(boxId,title, msg){
63         var box =document.getElementById(boxId);
64         box.childNodes[1].childNodes[1].innerHTML=title;
65         box.childNodes[3].innerHTML=msg;
66         box.style.display="inline";
67         pos();
68     }
69
70     function hideMsgBox(boxId){
71         hideShadow();
72         window.resizeTo(480, 480);
73     }
74     function showShadow(){
75         var shadow=document.querySelector("#shadow");
76         shadow.style.width= ""+document.documentElement.scrollWidth+"px";
77         if(document.documentElement.clientHeight>document.documentElement.scrollHeight)
78             shadow.style.height="" +document.documentElement.clientHeight+"px";
79         else
80             shadow.style.height="" +document.documentElement.scrollHeight+"px";
81         shadow.style.display="block";
82     }
83
84     function hideShadow(){
85         var shadow=document.querySelector("#shadow");
86         shadow.style.display="none";
87         var showBtn =document.querySelector("#msgbox1");
88         showBtn.style.display="none";
89     }
90
```

```
91     function pos(){
92         var msgbox=document.querySelector("#msgbox1");
93         msgbox.style.left = (document.documentElement.clientWidth/2-msgbox.clientWidth/2)+"
94             px"
95         msgbox.style.top = (document.documentElement.clientHeight/2-msgbox.clientHeight/2)+"
96             px"
97     }
98     window.onresize = function() {
99         setTimeout(function(){pos()}, 10);
100     }
101
102     function check(){
103         showShadow();
104         var name=document.querySelector("#name");
105         if(name.value==""){
106             showMsgBox('msgbox1', "错误信息", "姓名不能为空");
107         }
108         else{
109             showMsgBox('msgbox1', "姓名", name.value);
110         }
111     }
112 </script>
```

```
112 <div id="shadow">
113 </div>
114 <div id="msgbox1" class="msg msgbox">
115     <div class="msgheader msg">
116         <div class="msgtitle msg">这是标题</div>
117         <div class="msgclose msg" onclick="hideMsgBox('msgbox1')">X</div>
118     </div>
119     <div class="msgcontent msg">这是内容
120     </div>
121     <div class="btns">
122         <input type="button" class="ok" value="OK" onclick="hideMsgBox('msgbox1')">
123     </div>
124 </div>
125 姓名: <input type="text" id="name">
126 <input type="button" onclick="check()" value="check">
127 </body>
128 </html>
```

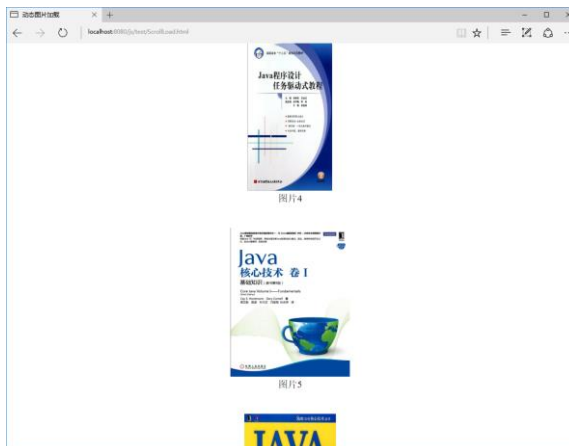
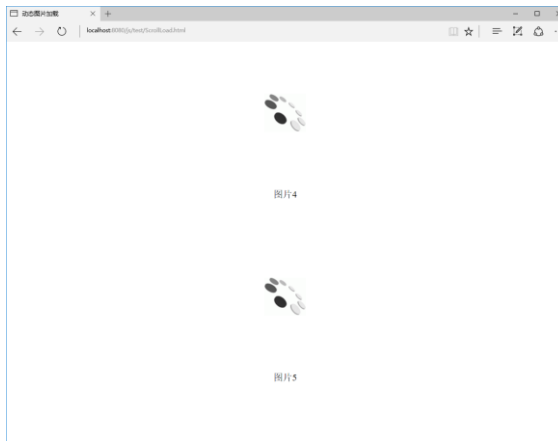
- 10、(ScrollLoad.html)为了节约流量，让网页中的图像只有出现在浏览器客户区时才显示，见 ScrollLoad.mp4。方法是用自定义属性 xSrc 保存图像的 URL，在网页滚动时对网页中所有图像(image 元素)进行判断，看看它们的上边沿或下边沿是否出现在浏览器客户区，如果是，则把 xSrc 的值拷贝到属性 Src 中（用 `getAttribute` 和 `setAttribute` 方法），此时浏览器才去下载并显示图像。 参考：Javascript(下)v2.pdf p53 p28-29



参考显示:

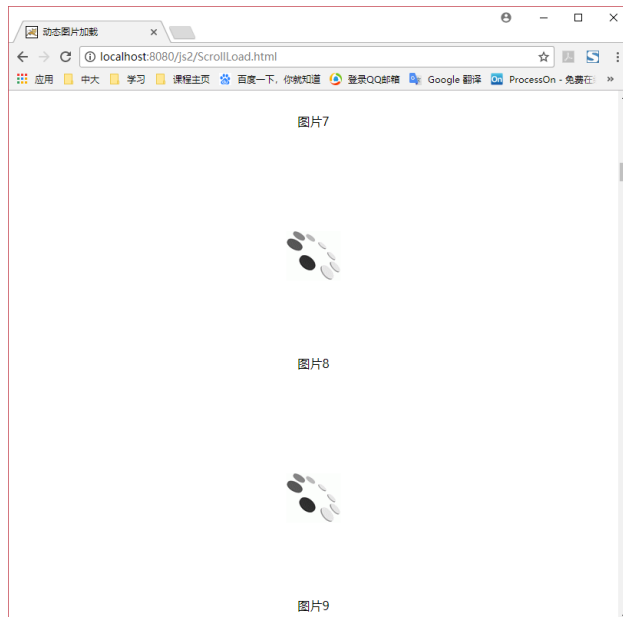


上滚之后:





完成实验后截屏：





源程序 (ScrollLoad.html):

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta content="text/html; charset=utf-8" /><title>动态图片加载</title>
5   <style>
6     body {text-align: center;}
7     .list {margin-bottom: 40px;}
8   </style>
9 </head>
10 <body>
11 <h1>动态图片加载</h1>
12 <div id="content">
13 </div>
14 <script type="text/javascript">
15   var allImgs = "";
16   var imgPath="http://localhost:8080/js2/imgs/";
17   for (var i = 1; i < 60; i++) {
18     allImgs += '<div class="list"><br /
      >图片' + i + '</div>';
21   }
22   document.getElementById("content").innerHTML = allImgs;
23   function windowHeight() {
24     return (document.compatMode=="CSS1Compat")?document.documentElement.clientHeight
25       :document.body.clientHeight;
26   }
27
28   function showImages() {
29     var docImg = document.images;
30     var cnt = docImg.length;
31     if (!cnt) return false;
32     for (var i = 0; i < cnt; i++) {
33       var o = docImg[i], imgHeight = 250;
34       // 新编代码加在这里
35       var top = o.getBoundingClientRect().top;
36       if((top + imgHeight) > 0 && (top + imgHeight) < windowHeight()){
37         o.setAttribute("src", o.getAttribute("xSrc"));
38       }
39     };
40   };
41   window.onscroll = function() {
42     setTimeout(function(){showImages()}, 1200);
43   }
44   showImages();
45 </script>
46 </body>
47 </html>
```

【完成情况】

是否完成了这些步骤? (√完成 ×未做 -未做完)

1 [√] 2 [√] 3 [√] 4 [√] 5 [√] 6 [√] 7 [√] 8 [√] 9 [√] 10 [√]

【实验体会】

写出实验过程中遇到的问题, 解决方法和自己的思考;并简述实验体会 (如果有的话)。

1. 在进行 menu.html 实验时发现自己的相对路径地址无法生效

这是由于在代码的开头定义了 base 基地址的属性, 所以如果我们使用相对路径就会以这个基地址为开头而形成绝对路径导致错误, 为了避免这个错误我们只有直接使用绝对路径才能实验成功。

```
<base href="http://www.sysu.edu.cn/2012/cn/">
```

2. 进行 timer1.html 实验时希望实现输出两位小数

在一开始进行输出时发现输出精度过高不符合我们的要求, 而为了结果这个问题, 实现固定输出, 可以使用 toFixed(int)函数。

```
timer.innerHTML = t.toFixed(2);
```



-
3. 进行 message.html 实验时希望实现当点击 OK 按钮后能够改变浏览器显示的大小
上网查阅资料后知道了 `window.resizeTo(int, int)` 函数可以直接对浏览器显示窗口大小进行改变，但是在一开始的使用时一直失败，无法正确实现这个功能，后来发现原来是浏览器的原因，用谷歌浏览器没办法使用这个功能，应该是没打开权限，转用 ie 之后就可以了。

```
window.resizeTo(480, 480);
```

【交实验报告】

(a) 每位同学单独完成本实验内容并填写实验报告上传。

(b) 截止日期: 2017 年 12 月 30 日 (周六) 23:00

上交作业网站: <http://172.18.187.11/netdisk/default.aspx?vm=15web>

文件夹: /实验上交/js2

上传文件: 学号_姓名_js2.doc (实验报告)

学号_姓名_js2.rar (所有源文件)