



院 系 数据科学与计算机学院 学 号 15352408 姓 名 张镓伟

【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

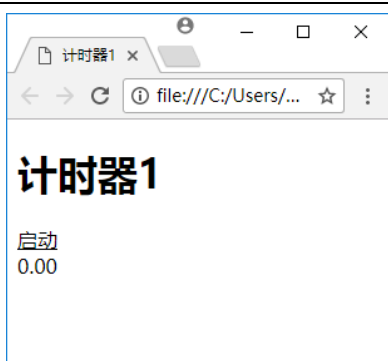
参考资料: JavaScript.ppt (下) 定时器

参考截屏:



视频演示见 timer1.mp4:

完成后截屏:

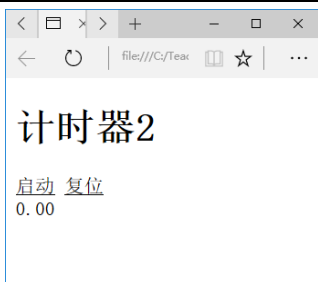


源码(timer1.html):

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
<title>计时器1</title>
<style>
  span {text-decoration:underline;}
  span:hover {cursor:pointer}
</style>
<script>
  var t = 0.00;
  var start = false;
  function timer(){
    document.getElementById('timer').innerHTML=t.toFixed(2);
    t=t+0.01;
    setTimeout("timer()",10);
  }
</script>
</head>
<body>
  <h1>计时器1</h1>
  <div><span id="start" onclick="timer()">启动</span></div>
  <div id="timer">0.00</div>
</body>
</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。

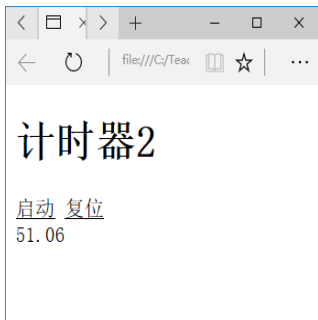
参考截屏:



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 == false) {
14         start = true;
15         document.getElementById('start').innerHTML="暂停";
16         timer();
17       } else {
18         start = false;
19         document.getElementById('start').innerHTML="启动";
20       }
21     }
```



```
22     function timer() {
23         if (start == true) {
24             document.getElementById('timer').innerHTML=t.toFixed(2);
25             t=t+0.01;
26             setTimeout("timer()",10);
27         }
28     }
29     function resetTimer() {
30         start = false;
31         t = 0.00;
32         document.getElementById('timer').innerHTML=t.toFixed(2);
33         document.getElementById('start').innerHTML="启动";
34     }
35 </script>
36 </head>
37 <body>
38     <h1>计时器2</h1>
39     <div><span id="start" onclick="toggleTimer()">启动</span> <span onclick="resetTimer()">复位</span></div>
40     <div id="timer">0.00</div>
41 </body>
42 </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。

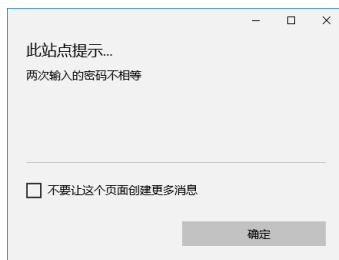
参考截屏：



输入 123 和 1234:



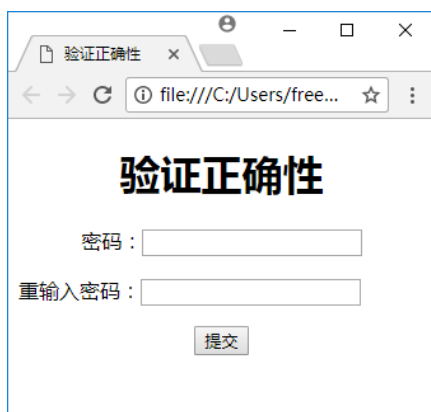
按提交按钮弹出信息，并取消了提交：



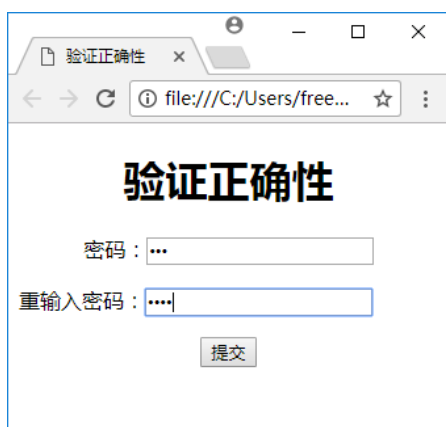
如果输入相同，则会提交，提交可能会出错（OK.jsp 不存在）。



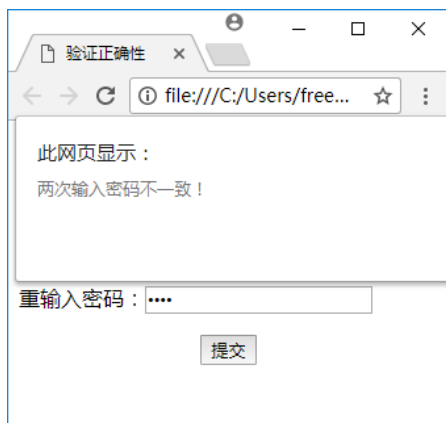
完成后截屏：



输入 123 和 1234：



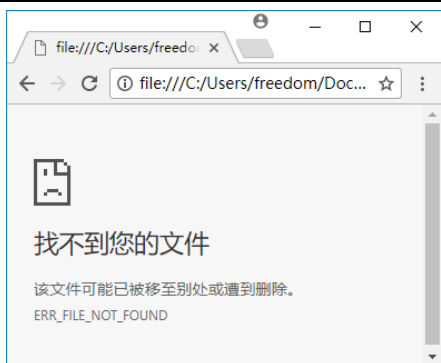
按提交按钮弹出信息，并取消了提交：



如果输入相同，则会提交，提交可能会出错（OK.jsp 不存在）：



实验报告



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:

开始界面:



自然滑动切换:



从第一张图点击右箭头切换到第二张图



还有点击左箭头可以回到前一张图以及点击右下角的点切换到对应图片的功能需要



直接打开网页尝试，截图无法体现。

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

Slider.html:

```
<!DOCTYPE html>
<html>
<head>
<title>Slider</title>
<link href="CSS/style.css" rel="stylesheet" type="text/css">
<script type="text/javascript" src="js/slider.js"></script>
</head>
<body onload="main()">
<div id="slideMain" onmouseover="MouseOver()" onmouseout="MouseOut()">
<div id="slides" class="slides">




</div>
<div id="point" class="point">



</div>
<div id="arrow1">

</div>
<div id="arrow2">

</div>
</div>
</body>
</html>
```

CSS:

```
#slideMain{
padding: 0px;
height: 300px;
width: 770px;
overflow: hidden;
position: relative;
margin: 50px auto;
}

.slides{
white-space: nowrap;
position: relative;
height: 300px;
width: 770px;
font-size: 0;
}

#point{
position: absolute;
height: 12px;
width: 50px;
bottom: 10px;
right: 50px;
white-space: nowrap;
overflow: hidden;
}

#point:hover{
cursor: pointer;
}
```



```
#p1{
  position: relative;
  top: -24px;
}
#p2{position: relative;}
#p3{position: relative;}

#arrow1{
  position: absolute;
  height: 80px;
  width: 32px;
  top: 125px;
  left: 10px;
  overflow: hidden;
  visibility: hidden;
}
#arrow11{
  position: relative;
  top: 0px;
  left: 0px;
}
#arrow11:hover{
  cursor: pointer;
}
#arrow2{
  position: absolute;
  height: 80px;
  width: 32px;
  top: 125px;
  right: 10px;
  overflow: hidden;
  visibility: hidden;
}
#arrow21{
  position: relative;
  top: -80px;
  left: 0px;
}
#arrow21:hover{
  cursor: pointer;
}
```

JavaScript:

```
function getstyle(sname) {
  return document.getElementById(sname).style
}

var move=true;
var count=77;
var index=0;
var left=0;
var id=0;
```



```
function SlideOneImg() {
    if (count < 77) {
        getstyle("slides").left = (left-10) + "px";
        left = left - 10;
        count = count + 1;
        setTimeout["SlideOneImg()", 15];
    }
    else if (left <= -2310){
        getstyle("slides").left = "0px";
        left = 0;
    }
}

function slide() {
    if (move) {
        count = 0;
        var p1, p2= null;
        if (index==3)p1 = getstyle("p1")
        else p1 = getstyle("p" + (index+1).toString())
        index = (index + 1) % 4;
        if (index==3)p2 = getstyle("p1")
        else p2 = getstyle("p" + (index+1).toString())
        SlideOneImg();
        p1.top = "0px";
        p2.top = "-24px";
        if (index==3) index = 0;
    }
    setTimeout("slide()", 3000);
}

function MouseOver() {
    move = false;
    getstyle("arrow1").visibility="visible";
    getstyle("arrow2").visibility="visible";
}

function MouseOut() {
    move = true;
    getstyle("arrow1").visibility="hidden";
    getstyle("arrow2").visibility="hidden";
}

function main() {
    setTimeout("slide()", 3000);
}
```

```
function click(idx) {
    var p1, p2 = null;
    count=77;
    p1 = getstyle("p" + (index+1).toString())
    index=idx;
    p2 = getstyle("p" + (index+1).toString())
    if (index == 0)left = 0;
    else if (index == 1)left = -10*77;
    else left = -10*77*2;
    getstyle("slides").left = left + "px";
    p1.top = "0px";
    p2.top = "-24px";
}

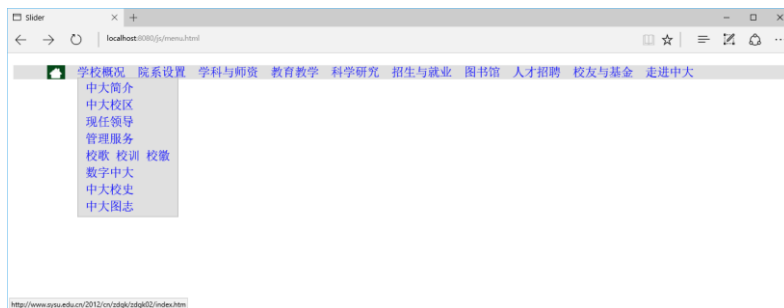
function click0() {click(0);}
function click1() {click(1);}
function click2() {click(2);}
```



```
function goLeft() {
    while (count < 77);
    p = getstyle("p" + (index+1).toString())
    p.top = "0px";
    index = (index - 1 + 3) % 3;
    p = getstyle("p" + (index+1).toString())
    p.top = "-24px";
    if (index == 0)left = 0;
    else if (index == 1)left = -10*77;
    else left = -10*77*2;
    getstyle("slides").left = left + "px";
}

function goRight() {
    while (count < 77);
    p = getstyle("p" + (index+1).toString())
    p.top = "0px";
    index = (index + 1) % 3;
    p = getstyle("p" + (index+1).toString())
    p.top = "-24px";
    if (index == 0)left = 0;
    else if (index == 1)left = -10*77;
    else left = -10*77*2;
    getstyle("slides").left = left + "px";
}
```

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：

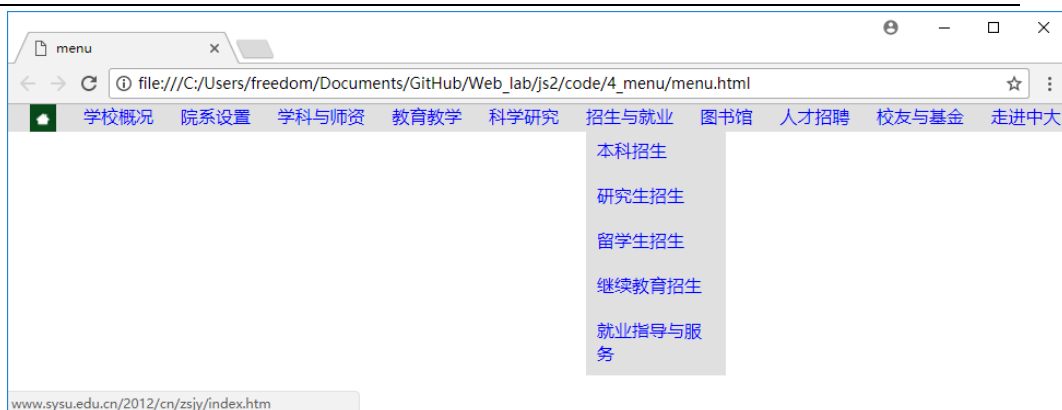
在不改动 html 的 body 内的已设计元素的情况下，如，不能为元素增加类和 id，实现下面功能：

- (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):

```
<!DOCTYPE html>
<html>
<head>
<title>menu</title>
<base href="http://www.sysu.edu.cn/2012/cn/">
<style type="text/css">
    *{ margin:0; padding: 0; list-style-type: none;}
    a:link,a:visited {text-decoration:none;color:blue}
    #nav>li ul li{display:inline-block; padding:10px;}
    #nav,#nav>li>ul {background: rgb(224,224,224);}
    #nav>li{
        position:relative;
        display:inline-block;
        margin-left:20px;
    }
    #nav>li>ul{
        position:absolute;
        width:8rem;
        left:0px;
        display: none;
    }
</style>
</head>
<body>
<ul id="nav">
<li>
    <a href="index.htm"><span class="home">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span></a>
</li>
<li><a href="zdgk/index.htm"><span>学校概况</span></a>
    <ul> <li><a href="zdgk/zdgk01/index.htm">中大简介</a></li>
        <li><a href="zdgk/zdgk02/index.htm">中大校区</a></li>
        <li><a href="zdgk/zdgk03/index.htm">现任领导</a></li>
        <li><a href="zdgk/zdgk04/index.htm">管理服务</a></li>
        <li><a href="zdgk/zdgk06/index.htm">校歌 校训 校徽</a></li>
        <li><a href="zdgk/zdgk07/index.htm">数字中大</a></li>
        <li><a href="zdgk/zdgk05/index.htm">中大校史</a></li>
        <li><a href="zdgk/zdgk08/index.htm">中大图志</a></li>
    </ul>
</li>
<li><a href="yx/index.htm"><span>院系设置</span></a>
</li>
<li><a href="xksz/index.htm"><span>学科与师资</span></a>
    <ul>
        <li><a href="xksz/xksz01/index.htm">学科建设</a></li>
        <li><a href="xksz/xksz02/index.htm">人才名录</a></li>
        <li><a href="xksz/xksz03/index.htm">名师垂范</a></li>
    </ul>
</li>
</ul>
```



```
<li><a href="jyjc/index.htm"><span>教育教学</span></a>
  <ul>
    <li><a href="jyjc/jyjc01/index.htm">本科教育</a></li>
    <li><a href="jyjc/jyjc02/index.htm">研究生教育</a></li>
    <li><a href="jyjc/jyjc03/index.htm">国际教育</a></li>
    <li><a href="jyjc/jyjc04/index.htm">继续教育</a></li>
  </ul>
</li>
<li><a href="kxyj/index.htm"><span>科学研究</span></a>
  <ul>
    <li><a href="kxyj/kxyj01/index.htm">重点研究机构</a></li>
    <li><a href="kxyj/kxyj04/index.htm">学术期刊</a></li>
    <li><a href="kxyj/kxyj05/index.htm">博士后科研流动站</a></li>
  </ul>
</li>
<li><a href="zszy/index.htm"><span>招生与就业</span></a>
  <ul>
    <li><a href="zszy/zszy01/index.htm">本科招生</a></li>
    <li><a href="zszy/zszy02/index.htm">研究生招生</a></li>
    <li><a href="zszy/zszy03/index.htm">留学生招生</a></li>
    <li><a href="jyjc/jyjc04/index.htm">继续教育招生</a></li>
    <li><a href="zszy/zszy04/index.htm">就业指导与服务</a></li>
  </ul>
</li>
<li><a href="http://library.sysu.edu.cn"><span>图书馆</span></a>
</li>
<li><a href="rczp/index.htm"><span>人才招聘</span></a>
</li>
<li><a href="xyjj/index.htm"><span>校友与基金</span></a>
</li>
<li><a href="zjzd/index.htm"><span>走进中大</span></a>
  <ul>
    <li><a href="http://myspace.sysu.edu.cn">5D空间</a>
    <li><a href="zjzd/zjzd02/index.htm">校区地图</a>
    <li><a href="zjzd/zjzd03/index.htm">生活服务</a>
    <li><a href="zjzd/zjzd04/index.htm">中大博物馆</a>
    <li><a href="zjzd/zjzd05/index.htm">校园文化</a>
  </ul>
</li>
</ul>
```

```
<script type="text/javascript">
var menu1=document.querySelectorAll("#nav>li");//取到一级菜单
var menu2=document.querySelectorAll("#nav>li>ul");//取到二级菜单（数组）
menu1[0].innerHTML='<a href="index.htm"><span></span></a>';
for(var i=0;i<11;i++){
  (function(i)
  {
    var f1 = function(){menu2[i].style.display="inline-block";};
    var f2 = function(){menu2[i].style.display="none";};
    menu2[i].parentNode.addEventListener("mouseover", f1, false);
    menu2[i].parentNode.addEventListener("mouseout", f2, false);
  })(i);
}
</script>
</body>
</html>
```

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):

```
<!DOCTYPE html>
<html>
<head>
<title>DOM</title>
<style>
  p { font-size:20px; }
</style>
</head>
<body>
<script>
  // 标题
  var h = document.createElement('h1');
  var txtH1 = document.createTextNode("静夜思");
  h.appendChild(txtH1);
  // 第一行
  var p1 = document.createElement('p');
  var txtP1 = document.createTextNode("床前明月光");
  var txtP2 = document.createTextNode("疑是地上霜");
  // 上标(1)
  var span1 = document.createElement('span');
  span1.style.verticalAlign = "super";
```



```
span1.style.fontSize = "12px";
var txtSup = document.createTextNode("(1)");
span1.appendChild(txtSup);
// 上标(2)
var span2 = span1.cloneNode(true);
span2.childNodes[0].nodeValue = "(2)";
// 逗号
var comma = document.createTextNode(", ");
// 句号
var period = document.createTextNode("。");
p1.appendChild(txtP1); p1.appendChild(span1); p1.appendChild(comma);
p1.appendChild(txtP2); p1.appendChild(span2); p1.appendChild(period);
// 第二行
var p2 = p1.cloneNode(true);
p2.childNodes[0].nodeValue = "举头望明月";
p2.childNodes[1].childNodes[0].nodeValue = "(3)";
p2.childNodes[3].nodeValue = "低头思故乡";
p2.childNodes[4].childNodes[0].nodeValue = "(4)";

document.body.appendChild(h);
document.body.appendChild(p1);
document.body.appendChild(p2);
</script>
</body>
</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:

Ajax-get

id:

[无]

Ajax-get

id:

[无]

(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：

Ajax-get

id:

正在查询...

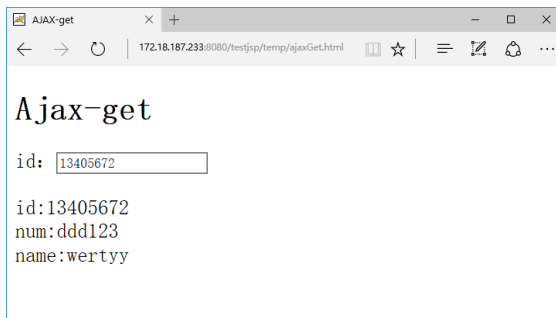
此站点提示...

id:13405672
num:dsd123
name:wertyy

确定



(3) 关闭弹出框后:

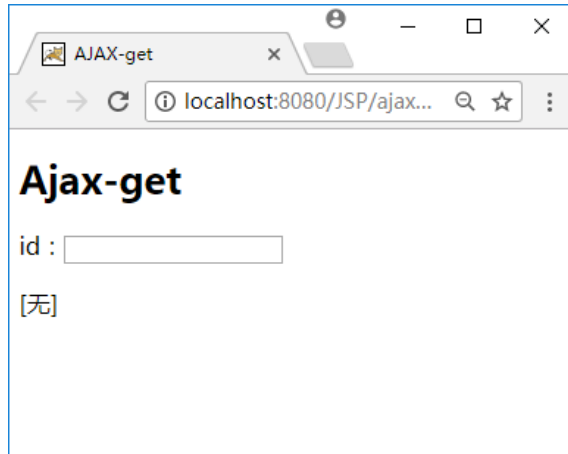


实验说明:

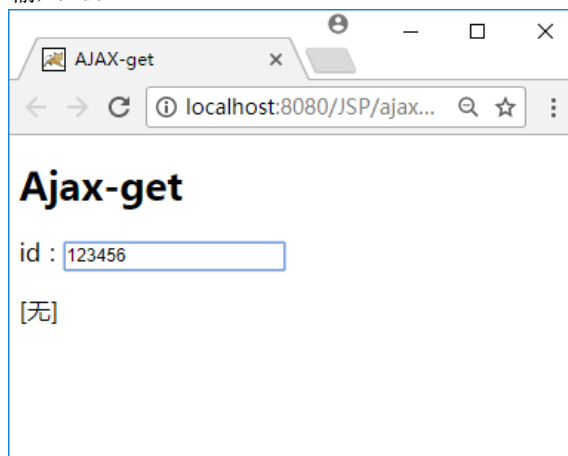
- (1) 用 <http://172.18.187.233:8080/jsp/browseStu.jsp> 查询数据。
MySQL 服务器: 172.18.187.233:53306 用户名:user 密码:123
数据库: teaching 表: stu 字段:id,num,name
- (2) 编写 get.jsp 程序可参考 JSP 实验(2)的 browseStu2.jsp, 在线调试时可以在文件名加上学号
get15000001.jsp, 以免冲突。
- (3) get.jsp 修改后有时要在浏览器中直接运行并刷新才生效。
例如: <http://172.18.187.233:8080/testjsp/temp/get.jsp>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

启动页面:

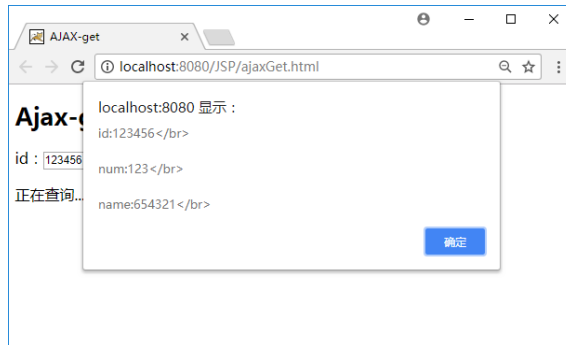


输入 id:

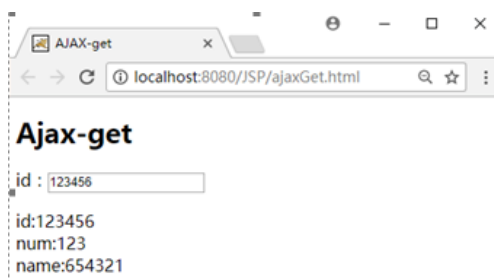




离开输入框之后:



最终结果显示



源码 ajaxGet.html 或 ajaxGet.jsp:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>AJAX-get</title>
  <style>
    p {font-size:20px}
  </style>
</head>
<body>
  <h1>Ajax-get</h1>
  <p>id: <input id="t1" type="text" onblur="query()"></p>
  <p id="t2">[无]</p>
  <script language="JavaScript" type="text/javascript">
    /*创建 XMLHttpRequest对象*/
    function createXMLHttpRequest(){
      var xmlhttp;
      if (window.XMLHttpRequest){
        xmlhttp = new XMLHttpRequest();
      }
      else{
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
      }
      return xmlhttp;
    }

    /* -----ajax方法-----*/
    function query(){
      var t1=document.getElementById("t1");

      xmlhttp=createXMLHttpRequest();
      if (xmlhttp==null){
        alert('您的浏览器不支持AJAX! ');
        return;
      }
      var msg="get.jsp?id="+t1.value;
      xmlhttp.open("get", msg, true);// 打开http请求 ( open) 的参数: get或post, un1, 是否异步发送
      xmlhttp.send(null);//发送http请求。 get只能用null作为参数
      var t2=document.getElementById("t2");
      t2.innerHTML="正在查询...";
      getReponse() //处理响应结果
    }
  </script>
</body>
</html>
```



源码 get.jsp:

7、用 ajax-post 技术(ajaxPost.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:




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 | 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:

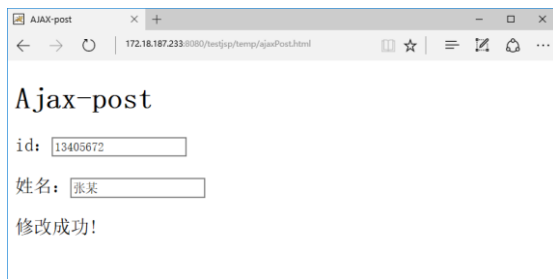


Ajax-post

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：

先浏览记录：

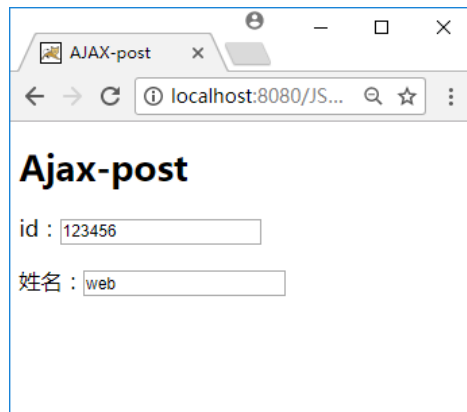


The screenshot shows a web browser window with the title '浏览学生名单'. The address bar shows the URL '172.18.187.233:8080/jsp/browseStu.jsp?pgno=0&pgcnt=4'. The main content area displays a table with student records. Below the table are links for '新增', '上一页', and '下一页'.

| id | 学号 | 姓名 | - |
|----------|-----------|--------|---------------------------------------|
| 123456 | 123 | 654321 | 修改 删除 |
| 13405703 | 111 | 陈某 | 修改 删除 |
| 15351234 | 123454321 | after | 修改 删除 |
| 15352029 | 15352029 | 粑粑 | 修改 删除 |

[新增](#) [上一页](#) [下一页](#)

选择输入第一个 id 并将姓名修改成新的：



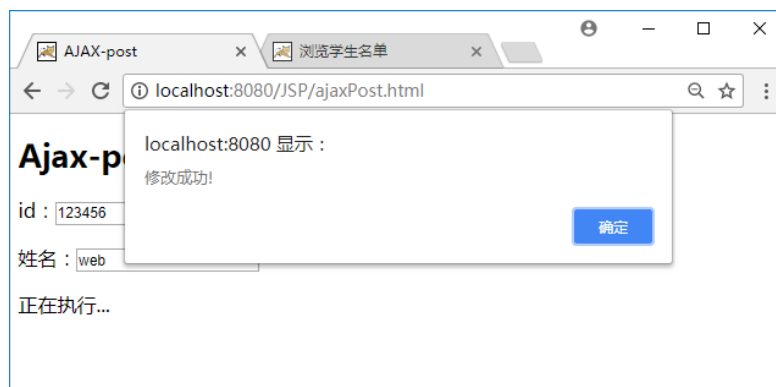
The screenshot shows a web browser window with the title 'Ajax-post'. The address bar shows the URL 'localhost:8080/JS...'. The form contains two input fields: 'id' with the value '123456' and '姓名' with the value 'web'.

Ajax-post

id :

姓名 :

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：



The screenshot shows the 'Ajax-post' form with a success message dialog box. The dialog box contains the text 'localhost:8080 显示 : 修改成功!' and a '确定' button. The form fields are 'id : 123456' and '姓名 : web'. Below the form, the text '正在执行...' is visible.

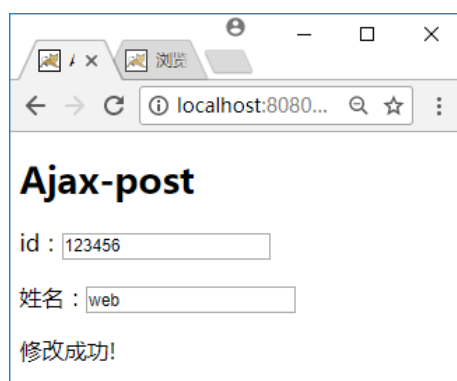
Ajax-p

localhost:8080 显示 :
修改成功!

id : 123456

姓名 : web

正在执行...



The screenshot shows the 'Ajax-post' form after a successful modification. The address bar shows the URL 'localhost:8080...'. The form contains two input fields: 'id' with the value '123456' and '姓名' with the value 'web'. Below the form, the text '修改成功!' is displayed.

Ajax-post

id :

姓名 :

修改成功!



查看网站上的记录的确被修改了：



源码(ajaxPost.html 或 ajaxPost.jsp):

```
<!DOCTYPE html>
<html>
<head>
  <title>AJAX-post</title>
  <style>
    p {font-size:20px;}
  </style>
</head>
<body>
  <h1>Ajax-post</h1>
  <p>id: <input id="t1" type="text"></p>
  <p>姓名: <input id="t2" type="text" onblur="update()"></p>
  <p><span id="t4"></span></p>
  <script language="JavaScript" type="text/javascript">
    /*创建 XMLHttpRequest对象*/
    function createXMLHttpRequest(){
      var xmlhttp;
      if (window.XMLHttpRequest){
        xmlhttp = new XMLHttpRequest();
      }
      else{
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
      }
      return xmlhttp;
    }
    /* -----ajax方法-----*/
    function update(){
      var t1=document.getElementById("t1");
      var t2=document.getElementById("t2");
      var t4=document.getElementById("t4");
      xmlhttp=createXMLHttpRequest();
      if (xmlhttp==null){
        alert('您的浏览器不支持AJAX! ');
        return;
      }
      //汉字需要编码
      var param="t1="+encodeURIComponent(t1.value)+"&t2="+encodeURIComponent(t2.value);
      xmlhttp.open("post","post.jsp",true);//(method,url: 文件在服务器上的位置,是否异步)
      xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
      xmlhttp.send(param);//将请求发送到服务器。string: 仅用于POST请求
      t4.innerHTML="正在执行...";
      getResponse() //处理响应结果
    }
  </script>

```



```
function getReponse(){
    xmlhttp.onreadystatechange = function(){
        if (xmlhttp.readyState == 4){
            if (xmlhttp.status >=200 && xmlhttp.status < 300 || xmlhttp.status == 304){
                alert(xmlhttp.responseText.replace(/(^\\s*)|(\\s*$)/g, ""));
                let t4=document.getElementById("t4");
                t4.innerHTML=xmlhttp.responseText;
            }
            else {
                alert("error")
                let t4 =document.getElementById("t4");
                t4.innerHTML="发生未知错误";
            }
        }
    }
}
</script>
</body>
</html>
```

源码(post.jsp):

```
<%@ page language="java" import="java.util.*,java.io.*,java.sql.*,java.awt.List"
    contentType="text/html; charset=utf-8"%>
<%request.setCharacterEncoding("utf-8");%>
<%!
    Connection conn;
    String msg=new String();
    //连接数据库
    boolean connect() {
        String connectString = "jdbc:mysql://172.18.187.233:53306/teaching"
            + "?autoReconnect=true&useUnicode=true&characterEncoding=UTF-8&useSSL=false";
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(connectString, "user", "123");
            return true;
        } catch (Exception e) {
            System.out.println(e.getMessage());
            msg = "连接不到数据库";
        }
        return false;
    }
    ResultSet executeUpdate(String sqlSentence) {
        Statement stat;
        ResultSet rs = null;
        try {
            stat =(Statement) conn.createStatement(); //获取执行sql语句的对象
            int cnt = stat.executeUpdate(sqlSentence);
            if(cnt!=0)msg="修改成功!";
            else msg="修改失败!";
        } catch (Exception e) {
            System.out.println(e.getMessage());
            msg="修改失败!";
        }
        return rs;
    }
}%>
```

```
<%
    String id=request.getParameter("t1");
    String name=request.getParameter("t2");
    if(connect()){
        msg="修改失败!";
        String sql=new String();
        sql="update stu set name='"+name+"' where id='"+id+"'";
        ResultSet rs = executeUpdate(sql);
    }
}%>
<%=msg%>
```



8、采用 div 实现以下富文本编辑。

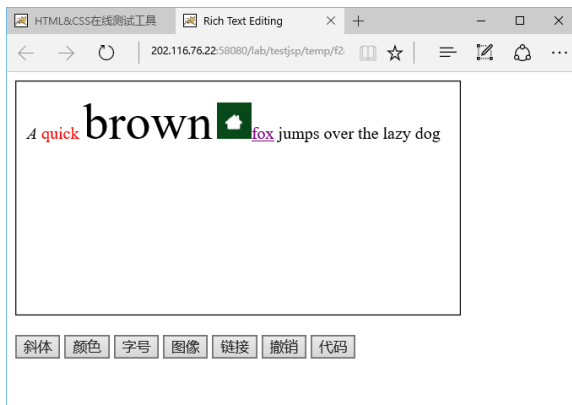
参考资料: Javascript(下)v2.pdf p49-p51 和附录 4

实验要求:

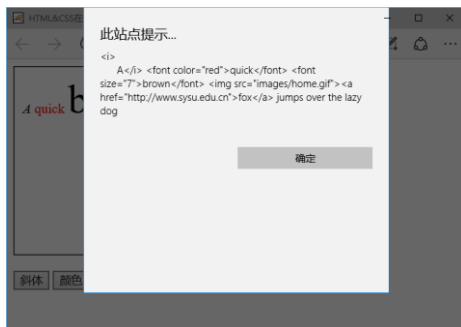
(1) 运行开始:



(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:

(1) 运行开始:

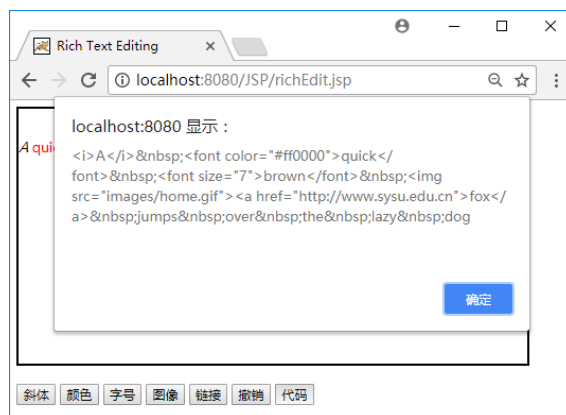




(2) 设置格式:



(3) 点击代码:



源码 richEdit.jsp:

```
<%@ page language="java" import="java.util.*,java.io.*"
contentType="text/html;charset=utf-8"%>
<!DOCTYPE html>
<html><head><title>Rich Text Editing</title></head>
<body>
<div id="richedit" name="edit" style="height:300px;width:600px;border:solid 2px black contenteditable="true">
</div><br>
<input type="button" value="斜体" onclick="italic()">
<input type="button" value="颜色" onclick="color()">
<input type="button" value="字号" onclick="big()">
<input type="button" value="图像" onclick="pic()">
<input type="button" value="链接" onclick="link()">
<input type="button" value="撤销" onclick="cancle()">
<input type="button" value="代码" onclick="show()">
<script>
function italic() {
    document.execCommand("italic",false,null);
}
function big() {
    document.execCommand("fontsize",false,7);
}
function color() {
    document.execCommand("forecolor",false,"red");
}
function pic() {
    document.execCommand("insertimage",false,"images/home.gif");
}
function link() {
    document.execCommand("createlink",false,"http://www.sysu.edu.cn");
}
function cancle() {
    var o = document.execCommand('undo');
}
function show() {
    var o=document.getElementById("richedit");
    alert(o.innerHTML);
}
</script>
</body>
</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。

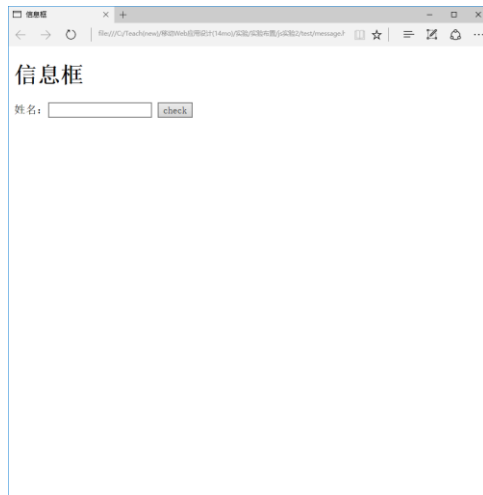
说明：

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

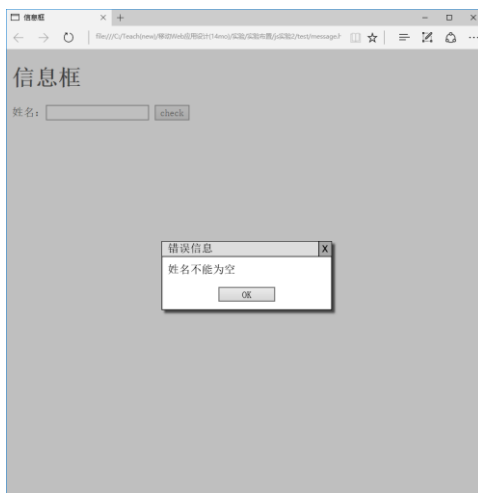
`document.documentElement.scrollHeight` 整个网页的高度（包含滚出客户区的部分）



参考显示：



点击 check 之后

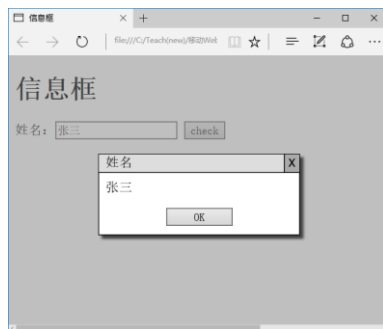


点击 OK，输入姓名：



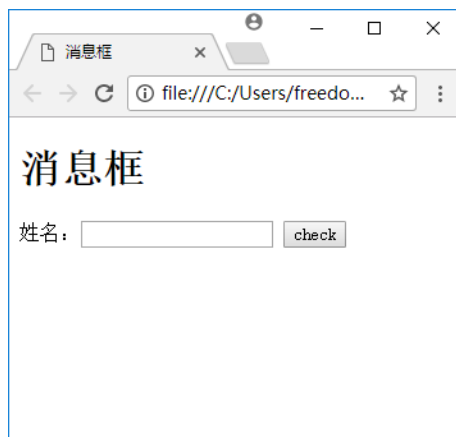
(窗口变小了)

点击 check 后:



完成实验后截屏:

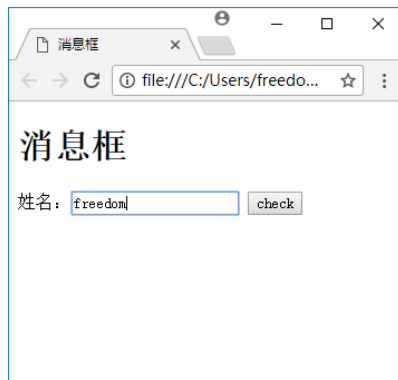
(1)启动窗口:



(2)姓名为空的提示:



(3)点击 X 或者 OK 之后输入名字:



(4)再次 check:



源程序 (message.html):

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
  <title>消息框</title>
  <link href="css/message.css" rel="stylesheet" type="text/css" charset="utf-8">
  <script type="text/javascript" src="js/message.js" charset="utf-8"></script>
</head>
<body>
  <h1>消息框</h1>
  <script type="text/javascript" src="js/message.js"></script>
  <div id="shadow">
  </div>
  <div id="msgbox1" class="msg msgbox">
    <div class="msgheader msg">
      <div class="msgtitle msg">这是标题</div>
      <div class="msgclose msg" onclick="hideMsgBox('msgbox1')">X</div>
    </div>
    <div class="msgcontent msg">这是内容</div>
    <div class="btns">
      <input type="button" class="ok" value="OK" onclick="hideMsgBox('msgbox1')">
    </div>
  </div>
  姓名: <input type="text" id="name">
  <input type="button" onclick="check()" value="check">
</body>
</html>
```

CSS:



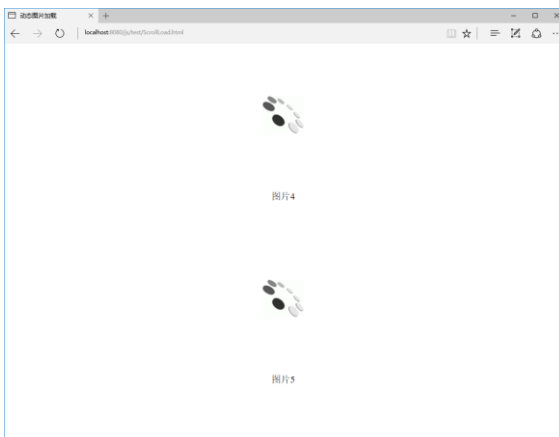
```
* {font-family:"宋体"}
#shadow {
    position: absolute;
    display: block;
    top: 0;
    left: 0;
    z-index: 10;
    background-color: #gray;
    opacity: 0.5;
}
#msgbox1 {
    display: none;
    position: fixed;
    width: 200px;
    height: 100px;
    border: solid 1px #black;
    background-color: #white;
    z-index: 20;
}
div.msgheader {
    background-color: #lightgray;
    border-bottom: solid 1px #black;
}
div.msgheader>.msg {
    padding: 4px;
    display: inline-block;
}
div.msgclose {
    border-left: solid 1px #black;
    background-color: #darkgray;
    float: right;
}
div.msgcontent { padding: 4px; }
div.btns {
    width: 80px;
    text-align: center;
    position: absolute;
    bottom: 10px;
    left: 50px;
}
div.btns>.ok {
    width: 100px;
    margin: auto;
}
```

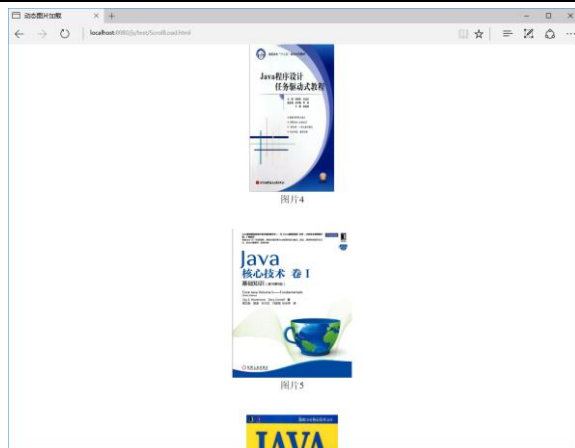
10、(ScrollLoad.html)为了节约流量，让网页中的图像只有出现在浏览器客户区时才显示，见 ScrollLoad.mp4。方法是用自定义属性 xSrc 保存图像的 URL，在网页滚动时对网页中所有图像(image 元素)进行判断，看看它们的上边沿或下边沿是否出现在浏览器客户区，如果是，则把 xSrc 的值拷贝到属性 Src 中（用 `getAttribute` 和 `setAttribute` 方法），此时浏览器才去下载并显示图像。 参考：Javascript(下)v2.pdf p53 p28-29

参考显示：



上滚之后：

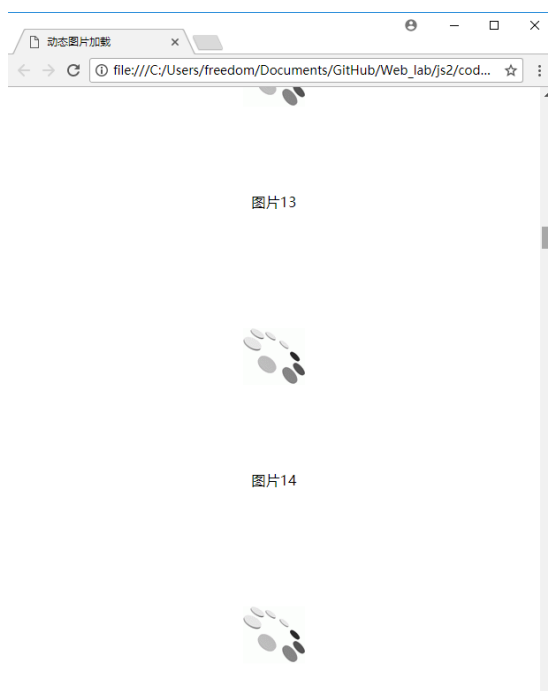




完成实验后截屏：



上滚之后：





源程序 (ScrollLoad.html):

```
<!DOCTYPE html>
<html>
<head>
  <meta content="text/html; charset=utf-8" /><title>动态图片加载</title>
  <style>
    body {text-align: center;}
    .list {margin-bottom: 40px;}
  </style>
</head>
<body>
  <h1>动态图片加载</h1>
  <div id="content">
  </div>
  <script type="text/javascript">
    var allImgs = "";
    var imgPath="imgs/";
    for (var i = 1; i < 60; i++) {
      allImgs += '<div class="list"><br />图片' + i + '</div>';
    }
    document.getElementById("content").innerHTML = allImgs;
    function windowHeight() { //获取页面浏览器客户区的高度
      return (document.compatMode=="CSS1Compat")?document.documentElement.clientHeight
      :document.body.clientHeight;
    }
    function showImages() {
      var docImg = document.images;
      var cnt = docImg.length;
      if (!cnt) return false;
      for (var i = 0; i < cnt; i++) {
        var o = docImg[i], imgHeight = 250;
        var clientHeight = windowHeight();
        var imgTop = o.getBoundingClientRect().top;
        if (imgTop+imgHeight >0 && imgTop + imgHeight < clientHeight){
          o.src=o.getAttribute("xSrc");
        }
      }
    };
    window.onscroll = function() {
      setTimeout(function(){showImages()}, 1200);
    }
    showImages();
  </script>
</body>
</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. 前两个实验很简单，第三个幻灯片实验要注意要实现循环滚动的效果，在第三张图片后面还要再加一张第一张图，当完成滑动到第 4 张图的时候，立马将第 4 张图切换到第一张图，视觉上并不会发现第 4 张和第 1 张图切换了，这样看上去就可以实现循环切换。

2. menu 设置了 base，所有外部文件会在 base 路径+相对路径下寻找，再不删除 base 的情况下我找不到本地的 home 图片，于是我找到了 base 地址里的 home 图片位置，让它加载那个路径下的图片，得以解决问题。

3. 后面的实验就更加熟悉了 js 和 ajax 的操作，同时还复习了一遍 css 布局。最后学会的动态加载图片的小技巧，很实用。

【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)