



院 系 数据科学与计算机学院 学 号 17341213 姓 名 郑康泽

【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);
}
```

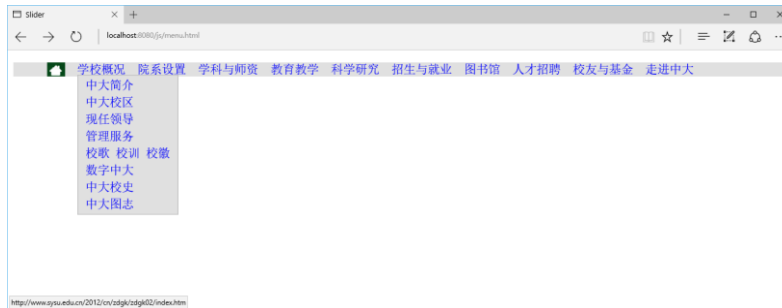
【实验环境】

- (1) 可以直接用浏览器打开 html 文件进行测试。
- (2) 在自己安装的 Web 服务器上测试。
- (3) 用自己安装的 Eclipse 或 intelij idea 调试
- (4) 在线调试: <http://103.26.79.35:8080/lab/js2/>

图像: <http://103.26.79.35:8080/lab/js2/imgs> (图片滚动, 最后一题)

【实验内容】

1、用 JavaScript 制作菜单:



参考资料: javascript(下).pdf

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):

```
<!DOCTYPE html>
<html>
<head>
  <title>menu</title>
  <base href="http://www.sysu.edu.cn/2012/cn/">
  <style>
    a:link, a:visited {
      text-decoration: none;
      color: blue
    }
    .home {
      background: url("F:/program/Web/js 实验 2/code/images/home.gif");
      display: inline-block;
      background-position-x: 30%;
      background-position-y: 60%;
      background-size: cover;
    }
    li {list-style-type: none;}
    ul#nav {
      background-color: rgb(224, 224, 224);
      height: 20px;
    }
    ul#nav>li {
      float: left;
      margin-right: 30px;
      position: relative;
      text-align: left;
    }
    ul#nav>li:nth-child(1) {text-align: center;}
    li>ul>li {
      width: 150px;
      text-align: left;
      background-color: rgb(224, 224, 224);
    }
    li>ul>li>a::before{content: "\00A0";}
    li>ul {
      position: absolute;
      left: -40px;
```



```
}
</style>
</head>
<body>
  <ul id="nav">
    <li><a href="index.htm"><span class="home">&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="xksj/index.htm"><span>学科与师资</span></a>
      <ul>
        <li><a href="xksj/xksj01/index.htm">学科建设</a></li>
        <li><a href="xksj/xksj02/index.htm">人才名录</a></li>
        <li><a href="xksj/xksj03/index.htm">名师垂范</a></li>
      </ul>
    </li>
    <li><a href="jyxx/index.htm"><span>教育教学</span></a>
      <ul>
        <li><a href="jyxx/jyxx01/index.htm">本科教育</a></li>
        <li><a href="jyxx/jyxx02/index.htm">研究生教育</a></li>
        <li><a href="jyxx/jyxx03/index.htm">国际教育</a></li>
        <li><a href="jyxx/jyxx04/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="zsjy/index.htm"><span>招生与就业</span></a>
      <ul>
        <li><a href="zsjy/zsjy01/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="jyxx/jyxx04/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>
var li_com = document.querySelectorAll("#nav>li");
for (var i=0; i<li_com.length; ++i) {
    if (li_com[i].childNodes.length == 4) {
        li_com[i].childNodes[2].style.visibility = "hidden";
    }
    (function(ind) {
        li_com[ind].addEventListener("mouseover", function() {
            if (li_com[ind].childNodes.length == 4) {
                li_com[ind].childNodes[2].style.visibility = "visible";
            }
        });
        li_com[ind].addEventListener("mouseout", function() {
            if (li_com[ind].childNodes.length == 4) {
                li_com[ind].childNodes[2].style.visibility = "hidden";
            }
        });
    })(i);
}
</script>
</body>
</html>
```

2、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.verticalAlign="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):

```
<!DOCTYPE html>
<html>
<head>
  <title>DOM</title>
  <style>
    p {font-size: 20px}
  </style>
  <script>
    window.onload = function() {
      var h1 = document.createElement("h1");
      var title = document.createTextNode("静夜思");
      h1.appendChild(title);

      var p1 = document.createElement("p");
      var txt1 = document.createTextNode("床前明月光");
      var sp1 = document.createElement("span");
```



```
sp1.style.verticalAlign = "super";
sp1.style.fontSize = "12px";
var txt2 = document.createTextNode("(1)");
sp1.appendChild(txt2);
var txt3 = document.createTextNode("， 疑是地上霜");
var sp2 = sp1.cloneNode(true);
sp2.childNodes[0].nodeValue = "(2)";
var txt4 = document.createTextNode("。");
p1.append(txt1);
p1.append(sp1);
p1.append(txt3);
p1.append(sp2);
p1.append(txt4);

var p2 = p1.cloneNode(true);
p2.childNodes[0].nodeValue = "举头望明月";
p2.childNodes[1].childNodes[0].nodeValue = "(3)";
p2.childNodes[2].nodeValue = "， 低头思故乡";
p2.childNodes[3].childNodes[0].nodeValue = "(4)";

document.body.append(h1);
document.body.append(p1);
document.body.append(p2);
}
</script>
</head>
<body>
</body>
</html>
```

3、用 ajax-get 技术实现查询功能。

参考资料: javascript(下).pdf AJAX 技术-GET

实验要求:

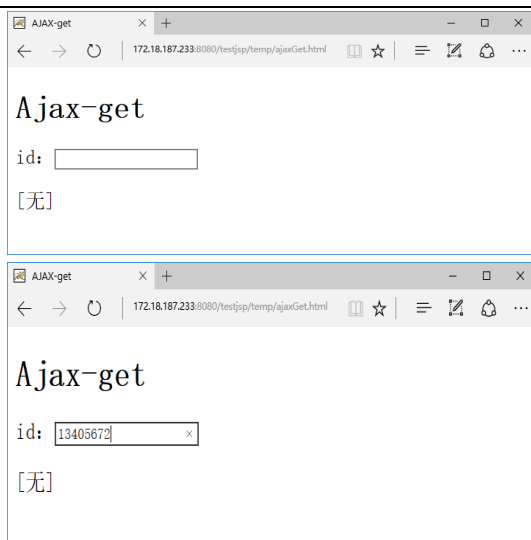
用 ajax-get 技术([ajaxGet.html](#) 和 [get.jsp](#))实现以下功能:

(1) 先浏览记录(<http://103.26.79.35:8080/jsp/browseStu.jsp>):

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](#)

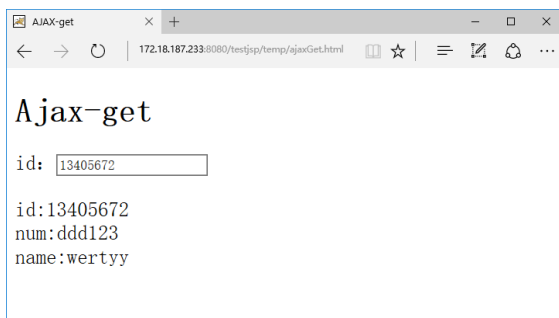
(2)在下面画面中输入 id(上面记录中有的):



- (3) 离开 id 输入域(采用 onblur 触发)后利用 ajax-get 技术调用 get.jsp
(http://103.26.79.35:8080/jsp/get.jsp?id=12347035) 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:

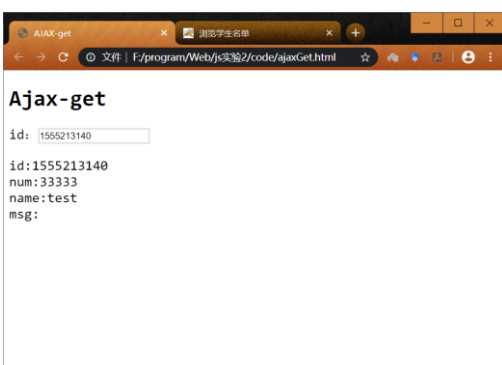
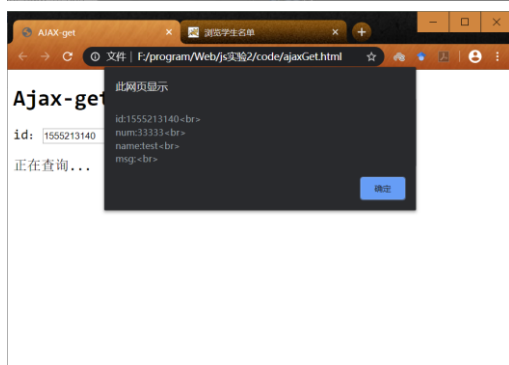
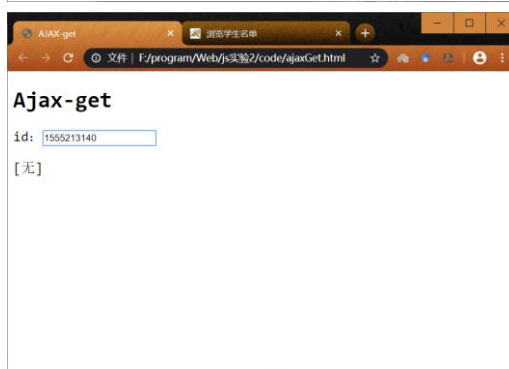
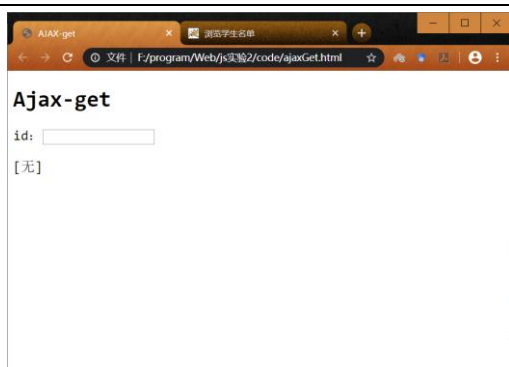


- (4) 关闭弹出框后:



完成后截屏:





源码 (ajaxGet.html) :

```
<!DOCTYPE html>
<html>
<head>
  <title>AJAX-get</title>
  <style>
    p {font-size:20px}
  </style>
</head>
<body>
```




```
<h1>Ajax-get</h1>
<p>id: <input id="t1" type="text" onblur="get_info()"></p>
<p id="t2">[无]</p>
<script>
    function get_info() {
        var p = document.getElementById("t2");
        p.innerHTML = "正在查询...";
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
            if (xmlhttp.readyState == 4) {
                if (xmlhttp.status >= 200 && xmlhttp.status < 300 || xmlhttp.status == 304) {
                    alert(xmlhttp.responseText);
                    p.innerHTML = xmlhttp.responseText;
                }
                else alert("error");
            }
        };
        xmlhttp.open("get", "http://103.26.79.35:8080/jsp/get.jsp?id=" +
document.getElementById("t1").value, true);
        xmlhttp.send(null);
    }
</script>
</body>
</html>
```

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

参考资料: Javascript 程序设计(下).pdf AJAX 技术-POST

实验要求:


(1) 先浏览记录(<http://103.26.79.35:8080/jsp/browseStu.jsp>):



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 (参数为 id 和 name)
(http://103.26.79.35:8080/jsp/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](#)

完成后截屏:

浏览学生名单

id	学号	姓名	-
1555213140	33333	test	修改 删除
1555213141	222222	改名字	修改 删除
1555213142	16349139	我是我	修改 删除
1555213143	4355	origin	修改 删除

[新增](#) [上一页](#) [下一页](#)

Ajax-post

id:

姓名:

修改成功!



id	学号	姓名	-
1555213140	33333	test	修改 删除
1555213141	222222	改名字	修改 删除
1555213142	16349139	我是我	修改 删除
1555213143	4355	modified	修改 删除

[新增](#) [上一组](#) [下一组](#)

源码 ajaxPost.html):

```
<!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="post_info()"></p>

    <p><span id="t4"></span></p>

    <script>

        function post_info() {

            var xmlhttp = new XMLHttpRequest();

            xmlhttp.onreadystatechange = function() {

                if (xmlhttp.readyState == 4) {

                    if (xmlhttp.status >= 200 && xmlhttp.status <= 300 || xmlhttp.status >= 304) {

                        document.getElementById("t4").innerHTML = "修改成功！ "

                    }

                }

            }

            var param = "id=" + document.getElementById("t1").value + "&name=" +

            encodeURIComponent(document.getElementById("t2").value);

            xmlhttp.open("post", "http://103.26.79.35:8080/jsp/post.jsp", true);

            xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded")

            xmlhttp.send(param);

        }

    </script>

</body>

</html>
```

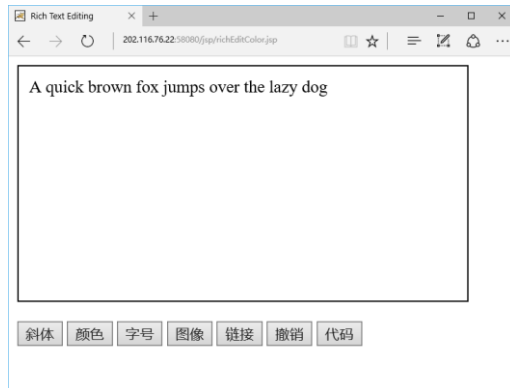


5、采用 div 实现以下富文本编辑。

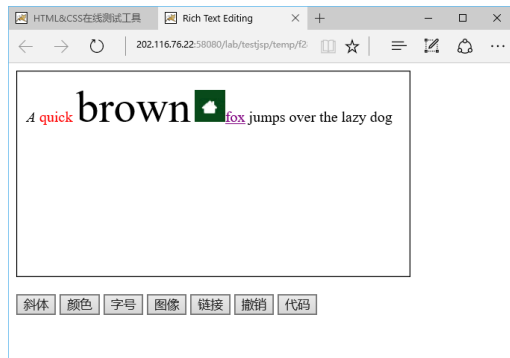
参考资料: Javascript(下).pdf 富文本编辑和附录 4

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:

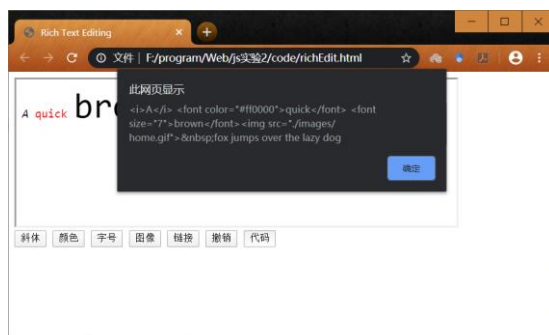




按撤销后, fox 的超链接消失



源码 richEdit.html:



6、(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。

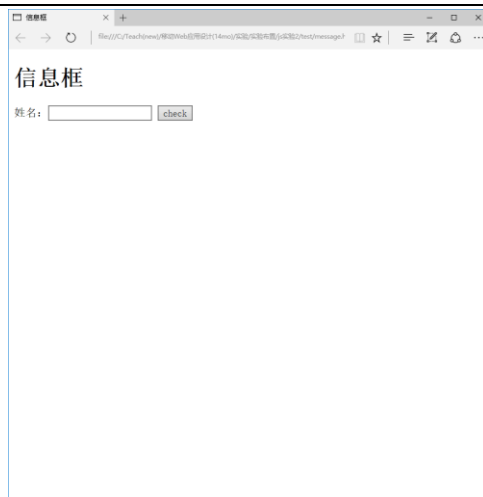
说明:

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

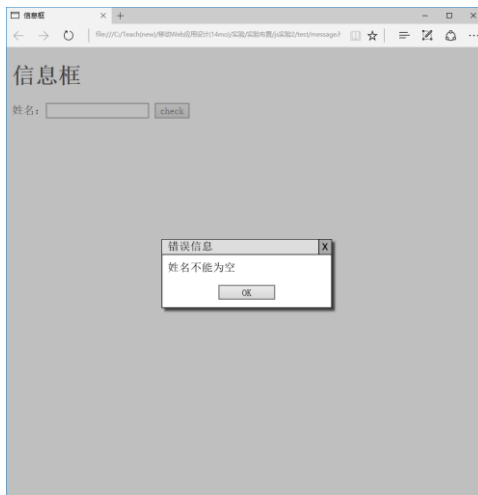
document.documentElement.scrollHeight 整个网页的高度(包含滚出客户区的部分)



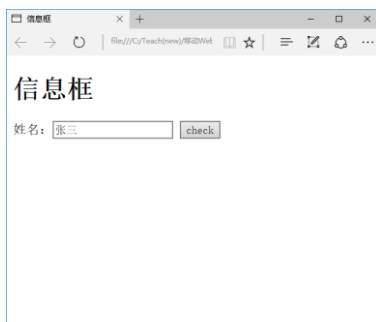
参考显示:



点击 check 之后

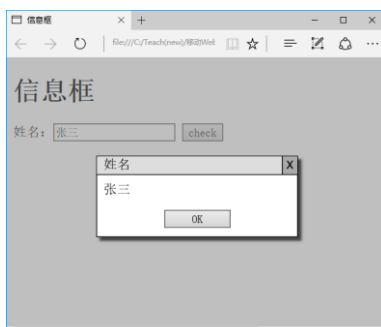


点击 OK，输入姓名：

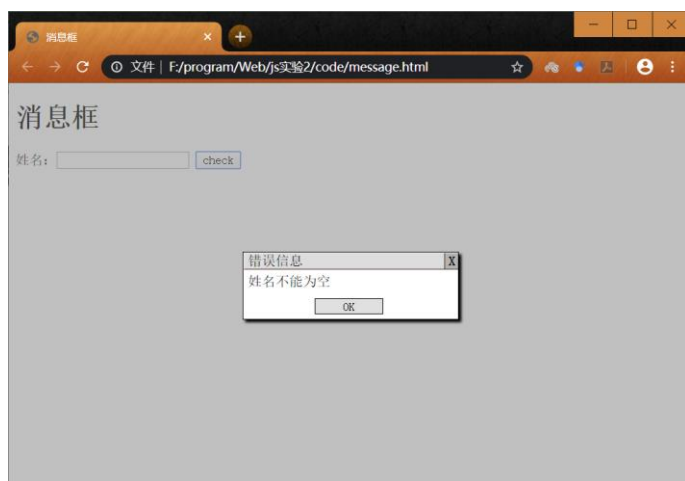
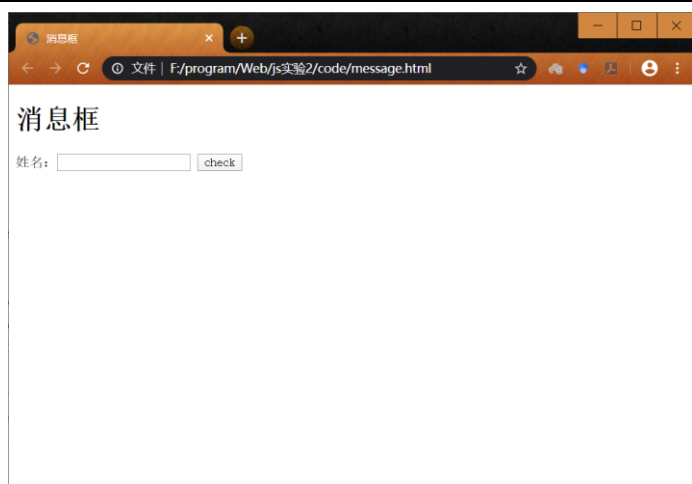


(窗口变小了)

点击 check 后：



完成实验后截屏：





源程序（message.html）：

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
  <title>消息框</title>
  <style>
    * {
      font-family: "宋体"
    }
    #msgbox1 {
      border: solid 1px black;
      box-shadow: 3px 3px 3px black;
      width: 250px;
      position: fixed;
      visibility: hidden;
      background-color: white;
      z-index: 3;
    }
    .msgtitle {
      border-bottom: solid 1px rgb(65, 24, 24);
      padding-left: 5px;
      background-color: rgb(221, 221, 221);
    }
    .msgclose {
      position: absolute;
      right: 0px;
      top: 0px;
      background-color: rgb(180, 180, 180);
      width: 15px;
      text-align: center;
      border-left: solid 1px rgb(65, 24, 24);
```




```
border-bottom: solid 1px rgb(65, 24, 24);
font-weight: bold;
cursor: default;
}
.msgcontent {
padding: 5px;
background-color: white;
}
.btns {
width: 85px;
margin: 5px auto;
}
.ok {
width: 80px;
background-color: rgb(224, 224, 224);
border: solid 1px black;
}
#shadow {
position: absolute;
top: 0;
left: 0;
z-index: 2;
background-color: gray;
opacity: 0.5;
visibility: hidden;
}
</style>
</head>
<body>
<h1>消息框</h1>
<script>
function showMsgBox(boxId, title, msg) {
let msgbox = document.getElementById(boxId);
// pos
msgbox.style.left = (document.documentElement.clientWidth / 2 - msgbox.clientWidth /
2) + "px";
msgbox.style.top = (document.documentElement.clientHeight / 2 - msgbox.clientHeight /
2) + "px";
// title msg
msgbox.childNodes[1].childNodes[1].childNodes[0].nodeValue = title;
msgbox.childNodes[3].childNodes[0].nodeValue = msg;
msgbox.style.visibility = "visible";
showShadow();
}
function hideMsgBox(boxId) {
let msgbox = document.getElementById(boxId);
```



```
        msgbox.style.visibility = "hidden";
        hideShadow();
    }
    function showShadow() {
        var shadow = document.getElementById("shadow");
        shadow.style.width = "" + document.documentElement.scrollWidth + "px";
        if (document.documentElement.clientHeight > document.documentElement.scrollHeight)
            shadow.style.height = "" + document.documentElement.clientHeight + "px";
        else
            shadow.style.height = "" + document.documentElement.scrollHeight + "px";
        shadow.style.visibility = "visible";
    }
    function hideShadow() {
        var shadow = document.getElementById("shadow");
        shadow.style.visibility = "hidden";
    }
    function check() {
        let name = document.getElementById("name");
        if (name.value == "")
            showMsgBox("msgbox1", "错误信息", "姓名不能为空");
        else
            showMsgBox("msgbox1", "姓名", name.value);
    }
</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>
```

7、(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：

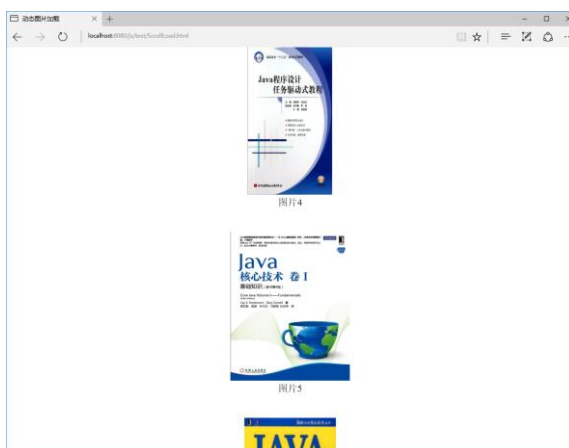
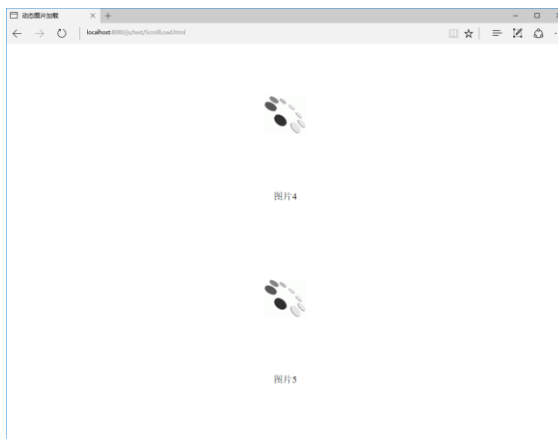


Javascript(下).pdf Scroll 与元素显示

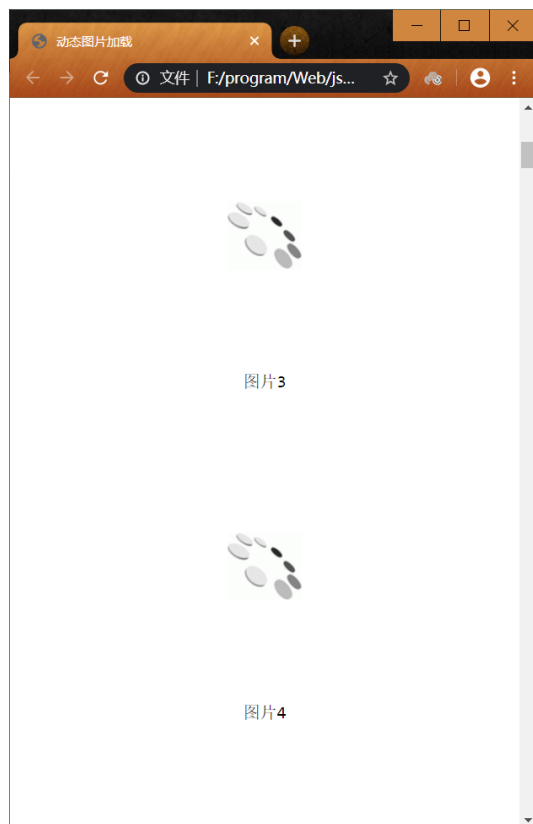
参考显示:



上滚之后:



完成实验后截屏:





图片3

图片4

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;
            if (o.getBoundingClientRect().top > 0 && o.getBoundingClientRect().top <
windowHeight() || o.getBoundingClientRect().bottom > 0 && o.getBoundingClientRect().bottom <
windowHeight())
                o.src = "./imgs/" + (i+1) + ".jpg";
        };
    };
    window.onscroll = function () {
        setTimeout(function () { showImages(); }, 1200);
    }
    showImages();
</script>
</body>
</html>
```

8、(timer.html)设计一个多计时器(秒表)的 Web 页，如下图所示，每次点击“增加计时器”按钮会增加一个计时器。

计时器

增加计时器(秒表)

3.48

启动

停止

暂停

17.11

启动

停止

暂停

3.72

启动

停止

暂停

16.67

启动

停止

暂停

5.86

启动

停止

暂停

7.25

启动

停止

暂停

16.05

启动

停止

暂停

9.63

启动

停止

暂停



(1) 每个计时器从 0 开始计时：

如果按了暂停键，则停止计时，此时只有按启动键才继续计时，按其他键无效；

如果按了停止键，则停止计时，此时只有按启动键才从 0 开始计时，按其他键无效；

具体操作见“timer.mp4”

(2) 尽量使用原型和闭包，例如，启动，停止和暂停都可以作为自定义的一个函数的原型的方法，每个计时器都是用这个函数生成的，它的按钮的事件处理程序中用闭包。

(3) 参考方法或函数：

```
document.querySelector();  
element.cloneNode(true);  
element.classList.remove();  
element.classList.add("pressed");  
setTimeout(timer, 10);  
var currentTime =(new Date()).getTime();
```

(4) 参考代码：

timer(参考代码).html

实现后截屏（四幅）：

按下“增加计时器”



按下“停止”



按下“启动”，重新开始计时



按下“暂停”



按下“停止”，继续计时



多个计时器



源程序（timer.html）：

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
  <title>计时器</title>
  <style>
    .time {
      display: inline-block;
      width: 3.6em;
      text-align: right;
      margin-right: 4px
    }
    #container,
    h1 {
      width: 20em;
      margin: 10px auto;
      text-align: center
    }
    .timer {
      margin-top: 20px
    }
    .pressed {
      background: yellow
    }
  </style>
  <script>
    function Timer() {
      Timer.prototype.start_func = function() {
        // alert(this);
        if(this.pause) {
          this.start = true;
          this.pause = false;
          this.start_time = (new Date).getTime();
          this.timer.childNodes[7].style.backgroundColor = "white";
        }
      }
    }
  </script>
</html>
```



```
else if(this.stop) {
    this.start = true;
    this.stop = false;
    this.start_time = (new Date).getTime();
    this.timer.childNodes[5].style.backgroundColor = "white";
}
this.timer.childNodes[1].childNodes[0].nodeValue = (((new Date).getTime() -
this.start_time) / 1000.00 + this.pass_time).toFixed(2);
this.timer.childNodes[3].style.backgroundColor = "yellow";
this.tid = setTimeout( this.start_func.bind(this), 10);
}
Timer.prototype.pause_func = function() {
    if (this.start) {
        this.start = false;
        this.pause = true;
        clearTimeout(this.tid);
        this.pass_time = Number(this.timer.childNodes[1].childNodes[0].nodeValue);
        this.timer.childNodes[3].style.backgroundColor = "white";
        this.timer.childNodes[7].style.backgroundColor = "yellow";
    }
}
Timer.prototype.stop_func = function() {
    if (this.start) {
        this.start = false;
        this.stop = true;
        clearTimeout(this.tid);
        this.timer.childNodes[3].style.backgroundColor = "white";
        this.timer.childNodes[5].style.backgroundColor = "yellow";
    }
}
function addTimer() {
    try{
        let t = new Timer();
        t.start = true;
        t.pause = false;
        t.stop = false;
        t.start_time = (new Date).getTime();
        t.pass_time = 0;

        t.timer = document.getElementById("container").childNodes[3].cloneNode(true);
        t.timer.style.display = "block";
        document.getElementById("container").appendChild(t.timer);

        t.tid = 0;
        t.start_func();
    }
```



```
t.timer.childNodes[3].addEventListener("click", t.start_func.bind(t));
t.timer.childNodes[5].addEventListener("click", t.stop_func.bind(t));
t.timer.childNodes[7].addEventListener("click", t.pause_func.bind(t));
    }catch(e) {alert(e);}
    }
</script>
</head>
<body>
    <h1>计时器</h1>
    <div id="container">
        <input type="button" class="add" value="增加计时器(秒表)" onclick="addTimer()">
        <div class="timer s0" style="display: none">
            <span class="time">0.00</span>
            <input type="button" class="start pressed" value="启动">
            <input type="button" class="stop" value="停止">
            <input type="button" class="pause" value="暂停">
        </div>
    </div>
</script>
</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 [√]

【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: 2020 年 7 月 19 日 (周日) 23:00

上交作业网站: <http://103.26.79.35/netdisk/default.aspx?vm=17web>

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)