**南京信息工程大学** Web技术与应用 **实验**(**实习**)**报告**

实验（实习）名称 **JavaScript编程2**  日期 2020.11.20 得分 指导教师 马瑞

系计算机与软件 专业 计算机科学与技术 年级18班次5姓名 申博文 学号201883290091

一、实验目的



**注：所有已给出的 截图在上交的实验报告中均删掉。**

二、 实验内容与步骤





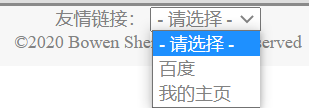


图 1 相关链接下拉框

主要代码如下：

    <div>友情链接：

      <select *id*="friendChains" *name*="friendChains" *onchange*="**jumpto**()">

        <option *value*="#">- 请选择 -</option>

        <option *value*="https://www.baidu.com">百度</option>

        <option *value*="https://sbwww.github.io">我的主页</option>

      </select>

    </div>

function **jumpto**() {

  location.href = document.**getElementById**("friendChains").value;

}





图 2 动态显示时间，每1000毫秒（1秒）刷新一次，时分秒有前导0

主要代码如下：

      <span *id*="timer"></span>

function **pad**(n) {

  if (n < 10) return "0" + n;

  return n;

}

function **timer**() {

  var date = new **Date**();

  var year = date.**getFullYear**();

  var month = date.**getMonth**() + 1;

  var day = date.**getDate**();

  var hour = date.**getHours**();

  var minute = date.**getMinutes**();

  var second = date.**getSeconds**();

  var time = "今天是" + year + "年" + month + "月" + day + "日 ";

  time += "北京时间：" + **pad**(hour) + ":" + **pad**(minute) + ":" + **pad**(second);

  document.**getElementById**("timer").innerHTML = time;

}

**setInterval**(function () {

**timer**();

}, 1000);



主要代码如下：

      <div *class*="imglist">

        <img *id*="autoPic" *src*="">

      </div>

function **autoPlayPics**() {

  var img = document.**getElementById**("autoPic");

  var index = 0;

  var prefix = "assets/images/";

  var imgs = new **Array**("ZL9012", "ZW9020", "ZW9021", "ZW0409B");

  var suffix = ".jpg";

**Silder** = () => {

    img.src = prefix + imgs[index] + suffix;

    index = (index + 1) % imgs.length;

  };

  var timer = **setInterval**(Silder, 1000);

  img.**onmouseover** = () => {

**clearInterval**(timer);

  };

  img.**onmouseout** = () => {

    timer = **setInterval**(Silder, 1000);

  };

}

window.**onload** = **autoPlayPics**();



图 3 图片轮播示例



代码：

    <div>

      <pre *id*="infoConfirm" *type*="pre"></pre>

    </div>

  username = document.**getElementById**("username").value;

  password = document.**getElementById**("password").value;

  confirm\_password = document.**getElementById**("confirm-password").value;

  age = document.**getElementById**("age").value;

  email = document.**getElementById**("email").value;

  occupation = document.**getElementById**("occupation").value;

  var degrees = document.**getElementsByName**("degrees");

  var skills = document.**getElementsByName**("skills");

  var send\_copy = document.**getElementsByName**("send-copy");

  var message = document.**getElementById**("message").value;

  var msg = "";

  msg += "Username: " + username;

  msg += "\nPassword: " + password;

  msg += "\nConfirm-password: " + confirm\_password;

  msg += "\nAge: " + age;

  msg += "\nEmail: " + email;

  msg += "\nOccupation: " + occupation;

  msg += "\nDegree: ";

  for (let index = 0; index < degrees.length; index++) {

    const degree = degrees[index];

    if (degree.checked) {

      msg += degree.value + " ";

    }

  }

  msg += "\nSkill: ";

  for (let index = 0; index < skills.length; index++) {

    const skill = skills[index];

    if (skill.checked) {

      msg += skill.value + " ";

    }

  }

  msg += "\nSend copy: ";

  for (let index = 0; index < send\_copy.length; index++) {

    const isSend = send\_copy[index];

    if (isSend.checked) {

      msg += isSend.value + " ";

    }

  }

  msg += "\nMessage: " + message + "\n";

  document.**getElementById**("infoConfirm").innerHTML = msg;

运行结果：



图 4 点击提交后，在页面下方显示表格内容



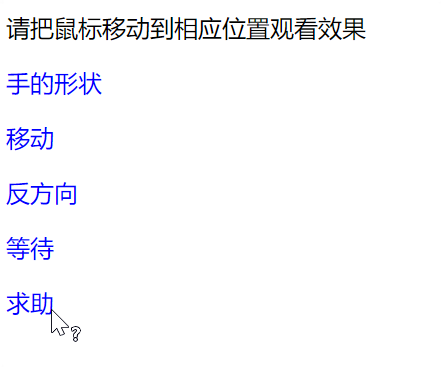
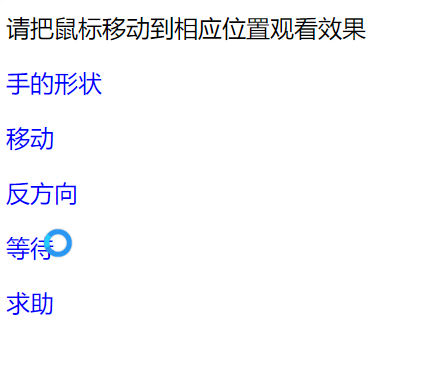
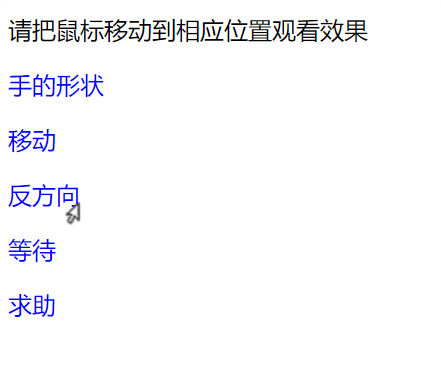
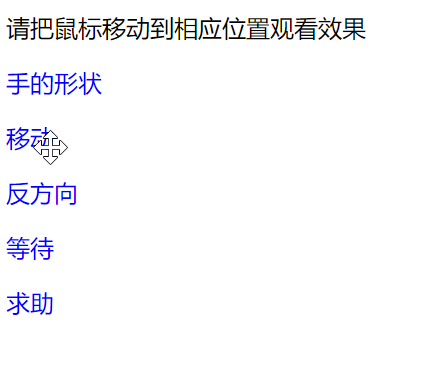
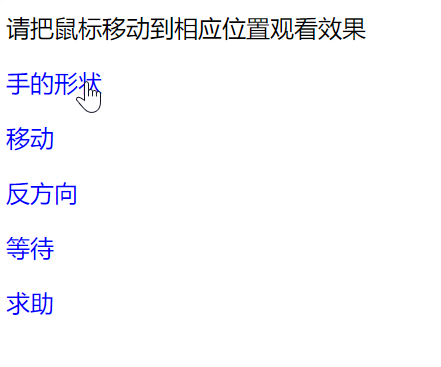


图 5 光标移动至不同文字上方的不同光标效果

主要代码如下：

    <div>

      <p>请把鼠标移动到相应位置观看效果</p>

    </div>

    <div>

      <font *color*="blue">

        <p *style*="cursor: pointer" *color*="green">手的形状</p>

        <p *style*="cursor: move">移动</p>

        <p *style*="cursor: url(assets/images/MirrorCursor\_16.png), auto">反方向</p>

        <p *style*="cursor: wait">等待</p>

        <p *style*="cursor: help">求助</p>

      </font>

    </div>





图 6 初始页面，默认数量为1，在页面加载时计算金额和总价



图 7 点击“+”号按钮后，数量加一，金额和总价对应地变化，也可以直接在文本框输入改变数量，文本框内容改变并失焦时触发onchange



图 8 勾选“全选”左侧的多选框，表格中的所有多选框被勾选，反之取消勾选



图 9 通过点击“-”号按钮，“删除”按钮，或在“数量”文本框输入0后，删除该商品

主要代码如下：

  <div *class*="content">

    <div *class*="logo">

      <img *src*="http://shopping.dangdang.com/shoppingcart/images/dd\_logo.jpg?v=20181226">

    </div>

    <div *class*="cartList" *id*="cartList">

      <table>

        <tr>

          <th><input *type*="checkbox" *name*="selectAll" *id*="selectAll" *onclick*="**selectAll**()">全选</th>

          <th>商品图片</th>

          <th>商品信息</th>

          <th>单价（元）</th>

          <th>数量</th>

          <th>金额（元）</th>

          <th>操作</th>

        </tr>

        <tr>

          <td><input *type*="checkbox" *name*="select"></td>

          <td>

            <img *src*="http://img3x9.ddimg.cn/88/32/28558519-1\_m\_6.jpg">

          </td>

          <td>Flutter开发实战详解</td>

          <td>¥<input *class*="price" *type*="text" *value*="85.40" *id*="price\_1">

          </td>

          <td>

            <input *class*="minus" *type*="button" *value*="-" *onclick*="**changeNum**(this)">

            <input *class*="amount" *type*="text" *value*="1" *onchange*="**changeNum**(this)">

            <input *class*="plus" *type*="button" *value*="+" *onclick*="**changeNum**(this)">

          </td>

          <td>¥<span *name*="sums">85.40</span></td>

          <td>

            <p>收藏</p>

            <p *onclick*="**deleteGood**(this)">删除</p>

          </td>

        </tr>

        <tr>

          <td><input *type*="checkbox" *name*="select"></td>

          <td>

            <img *src*="http://img3x7.ddimg.cn/6/24/28521807-1\_m\_9.jpg">

          </td>

          <td>Flutter实战</td>

          <td>

            ¥<input *class*="price" *type*="text" *value*="72.60" *id*="price\_2">

          </td>

          <td>

            <input *class*="minus" *type*="button" *value*="-" *onclick*="**changeNum**(this)">

            <input *class*="amount" *type*="text" *value*="1" *onchange*="**changeNum**(this)">

            <input *class*="plus" *type*="button" *value*="+" *onclick*="**changeNum**(this)">

          </td>

          <td>¥<span *name*="sums">72.60</span></td>

          <td>

            <p>收藏</p>

            <p *onclick*="**deleteGood**(this)">删除</p>

          </td>

        </tr>

        <tr>

          <td *colspan*="5"><input *class*="pay" *type*="button" *value*="结算" *onclick*="**total**()"></td>

          <td>¥<span *id*="totalPrice">0.00</span></td>

        </tr>

    </div>

  </div>

  <script *type*="text/javascript" *src*="assets/js/ex4\_4.js"></script>

function **changeNum**(node) {

  var d;

  var currentAmount;

  if (node.value == "-") {

    currentAmount = node.nextElementSibling;

    d = -1;

  } else if (node.value == "+") {

    currentAmount = node.previousElementSibling;

    d = 1;

  } else {

    currentAmount = node;

    d = 0;

  }

  currentAmount.value = **parseInt**(currentAmount.value) + d;

  if (currentAmount.value <= 0 || **isNaN**(currentAmount.value)) {

**deleteGood**(node);

  }

  var td = node.parentNode;

  price = **parseFloat**(

    td.previousElementSibling.**getElementsByTagName**("input")[0].value

  );

  amount = **parseInt**(currentAmount.value);

  td.nextElementSibling.childNodes[1].innerHTML = price \* amount;

**total**();

}

function **total**() {

  var sums = document.**getElementsByName**("sums");

  var totalPrice = 0;

  for (var i = 0; i < sums.length; ++i) {

    totalPrice += **parseFloat**(sums[i].innerHTML);

  }

  document.**getElementById**("totalPrice").innerHTML = totalPrice;

}

function **deleteGood**(node) {

  var tr = node.parentNode.parentNode;

  tr.parentNode.**removeChild**(tr);

**total**();

}

window.**onload**(**total**());

function **selectAll**() {

  var selectFlag = document.**getElementById**("selectAll").checked;

  var checkBoxes = document.**getElementsByName**("select");

  if (selectFlag) {

    checkBoxes.**forEach**((cb) => {

      cb.checked = 1;

    });

  } else {

    checkBoxes.**forEach**((cb) => {

      cb.checked = 0;

    });

  }

}

三、实验心得与体会

1. 在获取日期时，我发现月是从0开始，而日是从1开始，其原因是西方一般不是通过数字记录月份，而是使用January等词语表示，因此不是从1开始，而是根据计算机的习惯，从0开始。同理，在获取星期几的时候也是如此，从0开始，即Sunday开始；

2. 在设置光标样式时，并没有预设的样式是反方向光标，因此考虑使用cursor: url(), auto的方式，使用目标地址中的图片作为光标。

3. 访问兄弟节点可以使用nextElementSibling。nextSibling属性与nextElementSibling属性的差别在于，nextSibling属性返回元素节点之后的兄弟节点（包括文本节点、注释节点）；nextElementSibling属性只返回元素节点之后的兄弟元素节点（不包括文本节点、注释节点）。