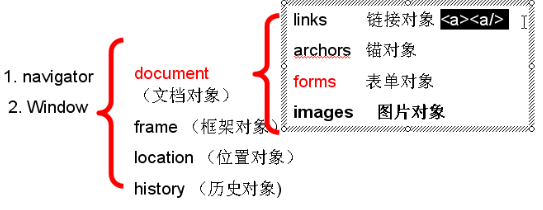
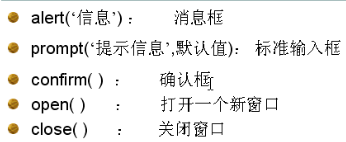
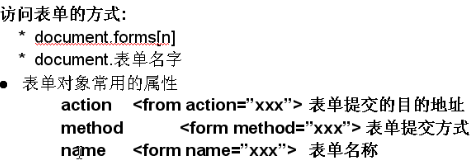
1. 20170520(黑马程序猿ajax和jquery基础入门)
   1. javascript回顾
      1. 浏览器对象树
         1. 
      2. window对象常用的方法
         1. 
      3. form表单对象
         1. 

|  |
| --- |
| //使用两种方式输出表单的action和method值 (使用表单的名称,使用表单数组forms) ----itcast20170522/webcontent/base/form1  //方法一  **var** formElement=document.form1;  alert(formElement.action);  alert(formElement.method);    //方法二  **var** formElement=document.forms[0];  alert(formElement.action);  alert(formElement.method); |

* + 1. Javascript定义函数的三种方式
       1. 打印和查询患者姓名--itcast20170522/webcontent/base/form1

|  |
| --- |
| <body>  <form name=*"form1"* action=*""* method=*"post"*>  患者姓名:<input type=*"text"* name=*"username"* value=*"zhang"* />  <input type=*"button"* name=*"ok"* value=*"打印患者信息"* onclick="printPerson()" />  <input type=*"button"* name=*"ok"* value=*"查询患者信息"* onclick="selectPerson()" />  </form>  </body>    <script language=*"JavaScript"*>  //通过javaScript函数的方式访问printPerson.html和selectPerson.html  **function** printPerson(){  **var** formElement=document.forms[0];  formElement.action="printPerson.html";  formElement.method="post";  formElement.submit();  }    **function** selectPerson(){  **var** formElement=document.forms[0];  formElement.action="selectPerson.html";  formElement.method="get";  formElement.submit();  }    </script> |

* + - 1. 构造函数的方法创建函数new Function（）--第一个字母大写

|  |
| --- |
| //方法二  //构造函数方式定义javascript函数，**前两个参数，后面是函数体**  **var** add=**new** Function('a','b','return a+b;');  //调用上面使用构造函数方式定义javascript函数  alert(add(3,4)); |

* + - 1. 函数直接量定义函数—可构造匿名函数itcast20170522/webcontent/base/functionmethod

|  |
| --- |
| //使用函数直接量的方式定义函数  **var** add=**function**(a,b){  **return** a+b;  }    //调用使用函数直接量定义的函数  alert(add(3,4)) |

* 1. Javascript基础加强—dom

DOM：DOM是document object model文档对象模型的缩写，根据w3c Dom规范，DOM是一种与浏览器，平台，语言无关的接口，使得你可以访问页面其他的标准组件

D：文档-html文档或xml文档

O：对象-document对象的属性和方法

M：模型

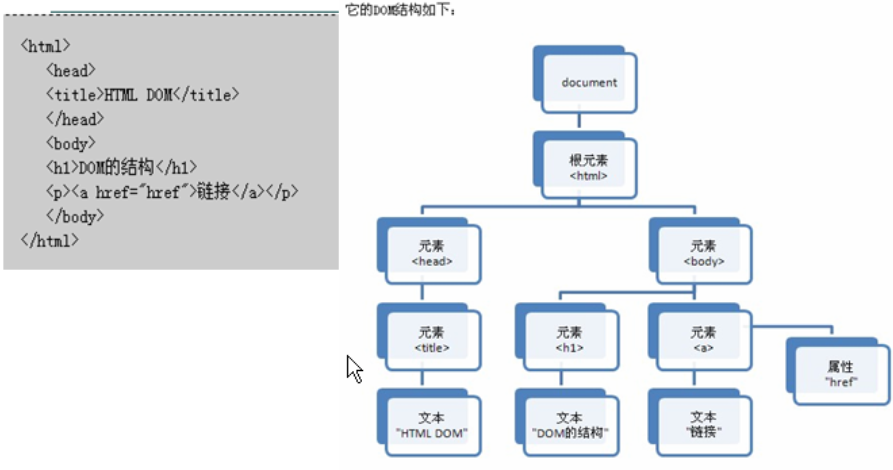
DOM是针对xml(HTML )的基于数的api

DOM树：节点的层次

DOM把一个文档表示为一棵家朴树（父，子，兄弟）

DOM定义了mode的接口以及许多节点类型来表示xml节点的多个方面

* + 1. dom结构



* + 1. Node接口的特性和方法



nodeValue只适用于文本节点和属性节点的值，对元素节点的值无效，只能用value进行获取

* + 1. 一些常用的函数
       1. getElementById

|  |
| --- |
| //输出 <input type="text" name="username" value="传智播客5周年\_1" id="tid" >标签value属性的值  **var** inputElement=document.getElementById("tid");  alert(inputElement.value);      //输出 <input type="text" name="username" value="传智播客5周年\_1" id="tid" >标签type属性的值  alert(inputElement.type); |

* + - 1. getElementsByName

|  |
| --- |
| var tnameElements=document.getElementsByName("tname");  for(var i=0;i<tnameElements.length;i++){  var tnameElement=tnameElements[i];  tnameElement.onchange=function(){  alert(this.value);  }  } |

* + - 1. getElementsByTagName

|  |
| --- |
| //输出所有下拉选 id="edu" 中option标签中value属性的值  /\*  \* <select name="edu" id="edu">  <option value="博士">博士^^^^^</option>  <option value="硕士">硕士^^^^^</option>  <option value="本科" selected="selected" >本科^^^^^</option>  <option value="幼儿园">幼儿园^^^^^</option>  </select>  \*/  // var eduElement=document.getElementById("edu");  // var optionElements=eduElement.getElementsByTagName("option");  // for(var i=0;i<optionElements.length;i++){  // //<option value="博士">博士^^^^^</option>  // var optionElement=optionElements[i]  // alert(optionElement.value);  // }      //输出所有下拉选内容  /\*  \* <select name="edu" id="edu">  <option value="博士">博士^^^^^</option>  <option value="硕士">硕士^^^^^</option>  <option value="本科" selected="selected" >本科^^^^^</option>  <option value="幼儿园">幼儿园^^^^^</option>  </select>    <select name="job" id="job" >  <option value="美容">美容^^^^^</option>  <option value="IT">IT^^^^^</option>  <option value="程序员">程序员^^^^^</option>  <option value="建筑师">建筑师^^^^^</option>  </select>  \*/  // var optionElements=document.getElementsByTagName("option");  // for(var i=0;i<optionElements.length;i++){  // //<option value="博士">博士^^^^^</option>  // var optionElement=optionElements[i]  // alert(optionElement.value);  // }    ////////////////////////////////////////////////////////////////////////////////////////////  //输出下拉选选中的值  /\*  \* <select name="edu" id="edu">  <option value="博士">博士^^^^^</option>  <option value="硕士">硕士^^^^^</option>  <option value="本科" selected="selected" >本科^^^^^</option>  <option value="幼儿园">幼儿园^^^^^</option>  </select>    <select name="job" id="job" >  <option value="美容">美容^^^^^</option>  <option value="IT">IT^^^^^</option>  <option value="程序员">程序员^^^^^</option>  <option value="建筑师">建筑师^^^^^</option>  </select>  \*/    alert(document.getElementById("edu").value);  alert(document.getElementById("job").value); |

* + - 1. hasChildNodes()

|  |
| --- |
| //查看id="tid\_1"的节点是否含有子节点  //<input type="text" name="tname" value="传智播客5周年\_1" id="tid\_1"><br>  var tidElement=document.getElementById("tid\_1");  alert(tidElement.hasChildNodes()); |

* + - 1. NodeName,nodeType,nodeValue

|  |
| --- |
| //元素节点 id="tid\_1" 输出nodeName nodeType nodeValue  // <input type="text" name="tname" value="传智播客5周年\_1" id="tid\_1" ><br>  // var tnameElement=document.getElementById("tid\_1");  // alert(tnameElement.nodeName); //input  // alert(tnameElement.nodeType); //1  // alert(tnameElement.nodeValue); //null    //文本节点 id="pid" 输出nodeName nodeType nodeValue  /\*  \* <p id="pid" name="8888">  明天上课  </p>  \*/  // var pidElement=document.getElementById("pid");  // //var textElement=pidElement.firstChild;  // //var textElement=pidElement.lastChild;  // var textElement=pidElement.childNodes[0];  //  // alert(textElement.nodeName); //#text  // alert(textElement.nodeType); //3  // alert(textElement.nodeValue); //明天上课  //属性节点 id="pid" 中name这个属性节点 输出nodeName nodeType nodeValue  /\*  \* <p id="pid" name="8888">  明天上课  </p>  \*/  var pidElement=document.getElementById("pid");  var nameElement=pidElement.getAttributeNode("name");  alert(nameElement.nodeName); //name  alert(nameElement.nodeType); //2  alert(nameElement.nodeValue); //8888 |

* + - 1. replaceChild()

|  |
| --- |
| //点击北京节点， 将被反恐节点替换  // document.getElementById("bj").onclick=function(){  // //获取北京节点  // var bjElement=document.getElementById("bj");  //  // //获取反恐节点  // var fkElement=document.getElementById("fk");  //  // //获取北京节点的父节点  // var bjParentElement=bjElement.parentNode;  //  // //替换  // var oldElement=bjParentElement.replaceChild(fkElement,bjElement);  //  // //测试  // alert(oldElement.getAttribute("value"));//返回的是已被替换的节点的指针（北京节点）  // } |

* + - 1. setAttribute()

|  |
| --- |
| //给这个节点增加 name属性 <li id="xj" value="xingji" name="xingjizhengba">星际争霸</li>  **var** xjElement=document.getElementById("xj");    //增加前获取属性  alert(xjElement.getAttribute("name"));    //增加属性  xjElement.setAttribute("name","xingjizhengba")    //增加后获取属性  alert(xjElement.getAttribute("name")); |

* + - 1. createElement(),createTextNode()

|  |
| --- |
| createElement    createTextNode    appendChild()    //增加城市节点 <li id="tj" v="tianjin">天津</li>  //<li></li>  **var** liElement=document.createElement("li");    //设置属性 <li id="tj" v="tianjin"></li>  liElement.setAttribute("id","tj");  liElement.setAttribute("v","tianjin");    //创建文本节点"天津"  **var** textElement=document.createTextNode("天津");    //增加文本节点"天津"到<li id="tj" v="tianjin"></li> 形成<li id="tj" v="tianjin">天津</li>  liElement.appendChild(textElement);    //获取父节点  **var** cityElement=document.getElementById("city");    //添加<li id="tj" v="tianjin">天津</li>到父节点下  cityElement.appendChild(liElement); |

* + - 1. insertBefore()和insertAfter()

|  |
| --- |
| insertBefore()    // 插入新的节点 <li id="tj" name="tianjin">天津</li>  // 在 <li id="cq" name="chongqing">重庆</li>的前面    //创建 <li id="tj" name="tianjin">天津</li>  **var** liElement=document.createElement("li");  liElement.setAttribute("id","tj");  liElement.setAttribute("name","tianjin");  **var** textElement=document.createTextNode("天津");  liElement.appendChild(textElement);    //获取重庆 <li id="cq" name="chongqing">重庆</li>  **var** cqElement=document.getElementById("cq");    //获取重庆节点的父节点  **var** cqParentElement=cqElement.parentNode;    //插入  cqParentElement.insertBefore(liElement,cqElement);  insertAfter()  // 插入新的节点 <li id="tj" name="tianjin">天津</li>  // 在 <li id="sh" name="shanghai">上海</li>的后面    //创建 <li id="tj" name="tianjin">天津</li>  **var** liElement=document.createElement("li");  liElement.setAttribute("id","tj");  liElement.setAttribute("name","tianjin");  **var** textElement=document.createTextNode("天津");  liElement.appendChild(textElement);    //获取上海节点  **var** shElement=document.getElementById("sh");    //获取上海节点的下一个兄弟节点 <li id="cq" name="chongqing">重庆</li>  **var** nextShElement=shElement.nextSibling;    //获取上海节点的父节点  **var** shParentElement=shElement.parentNode;    //添加到上海节点的下一个兄弟节点 <li id="cq" name="chongqing">重庆</li>的前面  shParentElement.insertBefore(liElement,nextShElement); |

* + - 1. removeChild()

|  |
| --- |
| //删除<ul id="city"> 下<li id="bj" name="beijing">北京</li>这个节点</ul>  //获取子节点  **var** bjElement=document.getElementById("bj");    //获取父节点  **var** cityElement=document.getElementById("city");    //删除  **var** oldElemnet=cityElement.removeChild(bjElement);    //测试  alert(oldElemnet.getAttribute("id")); |

* + - 1. 动态增加删除表格

|  |
| --- |
| window.onload函数是在dom结构绘制完毕或者，页面所有关联的文件必须加载完毕时候执行的  //dom结构绘制完毕,页面的所有关联的文件必须加载完毕(图片)  window.onload=**function**(){  //<button id="addUser">提交</button>  document.getElementById("addUser").onclick=**function**(){  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //获取文本框的值  /\*  \* 姓名: <input type="text" name="name" id="name" />&nbsp;&nbsp;  email: <input type="text" name="email" id="email" />&nbsp;&nbsp;  电话: <input type="text" name="tel" id="tel" /><br><br>  \*/    **var** nameValue=document.getElementById("name").value;  **var** emailValue=document.getElementById("email").value;  **var** telValue=document.getElementById("tel").value;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //创建td  /\*  \* <tr>  <td>Tom</td>  <td>tom@tom.com</td>  <td>5000</td>  <td><a href="deleteEmp?id=Tom">Delete</a></td>  </tr>  \*/  //创建nametd  **var** nameTd=document.createElement("td");  **var** nameText=document.createTextNode(nameValue);  nameTd.appendChild(nameText);    //创建emailtd  **var** emailTd=document.createElement("td");  **var** emailText=document.createTextNode(emailValue);  emailTd.appendChild(emailText);    //创建teltd  **var** telTd=document.createElement("td");  **var** telText=document.createTextNode(telValue);  telTd.appendChild(telText);    //创建atd  **var** aTd=document.createElement("td");    **var** aElement=document.createElement("a");  aElement.setAttribute("href","deleteEmp?id="+nameValue);  **var** deleteText=document.createTextNode("Delete");  aElement.appendChild(deleteText);  aTd.appendChild(aElement);  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //创建tr  **var** trElement=document.createElement("tr");    //增加td到tr上  trElement.appendChild(nameTd);  trElement.appendChild(emailTd);  trElement.appendChild(telTd);  trElement.appendChild(aTd);  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //增加tr到table上  //<table id="usertable" border="1" cellpadding="5" cellspacing=0>  //<tbody>  **var** tableElement=document.getElementById("usertable");    //创建tbody,为了跨浏览器,要创建tbody  **var** tbodyElement=document.createElement("tbody");  tbodyElement.appendChild(trElement);    //放置tbody到table上  tableElement.appendChild(tbodyElement);    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //删除  aElement.onclick=**function**(){  //使网页的链接失效  **return** delTr(aElement);  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  }  }  /\*\*  \* @param {Object} aElement  \*/  **function** delTr(aElement){  /\*  \* <tr>  <td>  Tom  </td>  <td>tom@tom.com</td>  <td>5000</td>  <td><a href="deleteEmp?id=Tom">Delete</a></td>  </tr>  \*/  **var** name=aElement.parentNode.parentNode.firstChild.firstChild.nodeValue;  //alert("name "+name)    **var** flag=window.confirm("您真的要删除["+name+"]这个用户吗?");  //alert(flag);    // 点击"取消" 按钮  **if**(!flag){  //使网页的链接失效  **return** **false**;  }    //删除  //获取tbody  **var** tbodyElement=aElement.parentNode.parentNode.parentNode;  //获取tr  **var** trElement=aElement.parentNode.parentNode;  //删除  tbodyElement.removeChild(trElement);  //使网页的链接失效    **return** **false**;  } |

* + - 1. 多选框的应用

|  |
| --- |
| window.onload=**function**(){  //<input type="button" name="checkall" id="checkall" value="全选" />  document.getElementById("checkall").onclick=**function**(){  **var** itemElements=document.getElementsByName("items");  **for**(**var** i=0;i<itemElements.length;i++){  //方法一: <input type="checkbox" name="items" value="足球" />足球  //itemElements[i].setAttribute("checked","checked");    //方法二  itemElements[i].checked="checked";  }  }    //<input type="button" name="checkall" id="checkallNo" value="全不选" />  document.getElementById("checkallNo").onclick=**function**(){  **var** itemElements=document.getElementsByName("items");  **for** (**var** i = 0; i < itemElements.length; i++) {  **var** itemElement=itemElements[i];    //方法一,IE行,其他的不行  //itemElement.setAttribute("checked",null);    //方法二: 都行  itemElement.checked=**null**;  }  }    //<input type="button" name="checkall" id="check\_revsern" value="反选" />  document.getElementById("check\_revsern").onclick=**function**(){  **var** itemElements=document.getElementsByName("items");  **for** (**var** i = 0; i < itemElements.length; i++) {  **var** itemElement = itemElements[i];    //itemElement.checked:如果选中为true,否则false  **if**(itemElement.checked){  itemElement.checked=**null**;  }**else**{  itemElement.checked="checked";  }  }  }  //<input type="checkbox" name="checkItems" id="checkItems" value="全选/全不选"/>全选/全不选  document.getElementById("checkItems").onclick=**function**(){  //alert(this.checked);  /\* var itemElements=document.getElementsByName("items");  for (var i = 0; i < itemElements.length; i++) {  var itemElement = itemElements[i];  if(this.checked){  itemElement.checked="checked";  }else{  itemElement.checked=null;  }  } \*/  **var** itemElements=document.getElementsByName("items");  **for** (**var** i = 0; i < itemElements.length; i++) {  **var** itemElement = itemElements[i];  itemElement.checked=**this**.checked;  }  }  } |
|  |

* + - 1. list的左移和右移

|  |
| --- |
| <table width=*"285"* height=*"169"* border=*"0"* align=*"left"* cellpadding=*"0"* cellspacing=*"0"* style="margin:*15px 0px 0px 15px*;">  <tr>  <td width=*"126"*>  <!--multiple="multiple" 能同时选择多个 size="10" 确定下拉选的长度-->  <select name=*"first"* multiple=*"multiple"* size=*10* class=*"td3"* id=*"s"*>  <option value=*"选项1"*>选项1</option>  <option value=*"选项2"*>选项2</option>  <option value=*"选项3"*>选项3</option>  <option value=*"选项4"*>选项4</option>  <option value=*"选项5"*>选项5</option>  <option value=*"选项6"*>选项6</option>  <option value=*"选项7"*>选项7</option>  <option value=*"选项8"*>选项8</option>  </select>  </td>  <td width=*"69"* valign=*"middle"*>  <input name=*"add"* id=*"add"* type=*"button"* class=*"button"* value=*"-->"* />  <input name=*"add\_all"* id=*"add\_all"* type=*"button"* class=*"button"* value=*"==>"* />  <input name=*"remove"* id=*"remove"* type=*"button"* class=*"button"* value=*"&lt;--"* />  <input name=*"remove\_all"* id=*"remove\_all"* type=*"button"* class=*"button"* value=*"&lt;=="* />  </td>  <td width=*"127"* align=*"left"*>  <select name=*"second"* size=*"10"* multiple=*"multiple"* class=*"td3"* id=*"second"*>  <option value=*"选项9"*>选项9</option>  </select>  </td>  </tr>  </table>  --------------------------------------------------------  window.onload=**function**(){  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //选中的从左端移到右端  //<input name="add" id="add" type="button" class="button" value="-->" />  document.getElementById("add").onclick=**function**(){  //获取<select name="first" multiple="multiple" size=10 class="td3" id="first">  **var** firstElement=document.getElementById("first");    //获取option  **var** optionElements=firstElement.getElementsByTagName("option");  **var** len=optionElements.length;  //alert(len)    //获取右边<select name="second" size="10" multiple="multiple" class="td3" id="second">  **var** secondElement=document.getElementById("second");    //遍历option  **for**(**var** i=0;i<len;i++){  **var** optionElement=optionElements[i];  /\*  \* firstElement.selectedIndex:  \* \* 表示当前选中的option的索引值,从0开始  \* \* 如果没有被选中的，selectedIndex=-1  \* \* 如果有多个被选中,selectedIndex的值永远是第一个被选中的值  \* \* javaScript的数组是动态的数组,数组的长度是可变的,索引值也在变  \*/    **if** (firstElement.selectedIndex != -1) {  alert(firstElement.selectedIndex);  secondElement.appendChild(optionElements[firstElement.selectedIndex]);  }  }  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //全部从左端移到右端  //<input name="add\_all" id="add\_all" type="button" class="button" value="==>" />  document.getElementById("add\_all").onclick=**function**(){  **var** firstElement=document.getElementById("first");    //获取option  **var** optionElements=firstElement.getElementsByTagName("option");  **var** len=optionElements.length;  //alert(len)    //获取右边<select name="second" size="10" multiple="multiple" class="td3" id="second">  **var** secondElement=document.getElementById("second");  **for** (**var** i = 0; i < len; i++) {  secondElement.appendChild(optionElements[0]);  }  }  //简化的  // document.getElementById("add\_all").onclick=function(){  // var firstElement=document.getElementById("first");  // var len=firstElement.length;  // //alert(len)  // var secondElement=document.getElementById("second");  // for (var i = 0; i < len; i++) {  // secondElement.appendChild(firstElement[0]);  // }  // }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //双击的从左端移到右端  // document.getElementById("first").ondblclick=function(){  // var firstElement=document.getElementById("first");  // var optionElements=firstElement.getElementsByTagName("option");  //  // var secondElement=document.getElementById("second");  // for(var i=0;i<optionElements.length;i++){  // secondElement.appendChild(optionElements[firstElement.selectedIndex]);  // }  // }    //优化的  document.getElementById("first").ondblclick=**function**(){  **var** secondElement=document.getElementById("second");  secondElement.appendChild(**this**[**this**.selectedIndex]);  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //选中的从右端移到左端  //<input name="remove" id="remove" type="button" class="button" value="&lt;--" />  document.getElementById("remove").onclick=**function**(){  **var** secondElement=document.getElementById("second");  **var** optionElements=secondElement.getElementsByTagName("option");  **var** len=optionElements.length;    **var** firstElement=document.getElementById("first");    **for**(**var** i=0;i<len;i++){  **if**(secondElement.selectedIndex!=-1){  firstElement.appendChild(optionElements[secondElement.selectedIndex]);  }  }  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //全部的从右端移到左端  //<input name="remove\_all" id="remove\_all" type="button" class="button" value="&lt;==" />  document.getElementById("remove\_all").onclick=**function**(){  **var** secondElement=document.getElementById("second");  **var** optionElements=secondElement.getElementsByTagName("option");  **var** len=optionElements.length;  **var** firstElement=document.getElementById("first");  **for**(**var** i=0;i<len;i++){  firstElement.appendChild(optionElements[0]);  }  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //双击的从右端移到左端  document.getElementById("second").ondblclick=**function**(){  **var** firstElement=document.getElementById("first");  firstElement.appendChild(**this**[**this**.selectedIndex]);  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  } |

* + - 1. 二级菜单联动（城市）

|  |
| --- |
| 解析xml  **function** parseXML(fileXmlName){  **var** xmlDoc;  **try** {//Internet Explorer 创建一个空的xml文档  xmlDoc = **new** ActiveXObject("Microsoft.XMLDOM");  }  **catch** (e) {  **try** {//Firefox, Mozilla, Opera, 创建一个空的xml文档  xmlDoc = document.implementation.createDocument("", "", **null**);  }  **catch** (e) {  }  }  //关闭异步加载  xmlDoc.async = **false**;  //加载xml文件  xmlDoc.load(fileXmlName);  **return** xmlDoc;  }  window.onload=**function**(){  document.getElementById("province").onchange=**function**(){  **var** pvalue=**this**.value;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //删除  /\*  \* <select id="city" name="city">  <option value="">请选择.....</option>  </select>  \*/  **var** cityElement=document.getElementById("city");  **var** oldOptionElements=cityElement.getElementsByTagName("option");  **for**(**var** i=oldOptionElements.length-1;i>0;i--){  cityElement.removeChild(oldOptionElements[i]);  }    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  **var** xmlDoc=parseXML("cities.xml");  **var** provinceXmlElements=xmlDoc.getElementsByTagName("province");  **var** provinceElement=**null**;  //alert(provinceXmlElements.length);  **for**(**var** i=0;i<provinceXmlElements.length;i++){    //<province name="吉林省">  **var** provinceXmlElement=provinceXmlElements[i];    //<province name="吉林省">  **var** nameAttr=provinceXmlElement.getAttribute("name");  **if**(nameAttr==pvalue){  //保存省的元素为全局变量  provinceElement=provinceXmlElement;  }  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  **if**(provinceElement!=**null**){  /\*  \* <province name="吉林省">  <city>长春</city>  <city>吉林市</city>  <city>四平</city>  <city>松原</city>  <city>通化</city>  </province>  \*/  **var** cityXmlElements=provinceElement.getElementsByTagName("city");  **for**(**var** i=0;i<cityXmlElements.length;i++){  **var** cityValue=cityXmlElements[i].firstChild.nodeValue;    **var** optionElement=document.createElement("option");  optionElement.setAttribute("cityValue",cityValue);  **var** optionTextElement=document.createTextNode(cityValue);  optionElement.appendChild(optionTextElement);    cityElement.appendChild(optionElement);    }  }  }  } |