1. **AJAX(Asynchronous JavaScript and XML)异步的Javascript与XML**
2. **异步指的是不用等到前一个操作完成就可进入下一个操作。俩个操作或多个操作可“同时”进行。这极大的提高了用户体验。**
3. **AJAX发送请求时并不刷新整个页面，只是进行局部刷新，因此我们并不能感知请求是否发送，只能通过第三方的调试工具才能看到，例如firebug等。**
4. **不同的浏览器对Ajax的实现方式是不同的，Ajax中一个重要的对象为XMLHttpRequest，其中微软的IE6对该对象的实现是采用ActiveXObject控件的方式，而其它浏览器则采取了内置实现的方式。因此在Ajax编程时必须对其进行区分。例如如下的ajax.jsp页面：**
5. <script type=*"text/javascript"*>
7. **var** xmlHttpRequest = **null**;//声明一个空对象以接收XMLHttpRequest对象
9. **function** **ajaxSubmit**(){
11. //alert(window.ActiveXObject);
13. **if**(window.ActiveXObject) {
15. xmlHttpRequest = **new** ActiveXObject("Microsoft.XMLHTTP");//IE浏览器
17. }**else** **if**(window.XMLHttpRequest) {
19. xmlHttpRequest = **new** XMLHttpRequest();//菲IE的其它浏览器
20. }
22. **if**(**null** != xmlHttpRequest) {
24. **var** v1 = document.getElementById("value1ID").value;
26. **var** v2 = document.getElementById("value2ID").value;
28. xmlHttpRequest.open("POST","AjaxServlet",**true**);//准备发送数据
30. xmlHttpRequest.onreadystatechange = ajaxCallback;//关联好Ajax的回调函数
32. //使用POST方式提交，必须加上如下一行
33. xmlHttpRequest.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
35. xmlHttpRequest.send("v1=" + v1 + "&v2=" + v2);//真正向服务器发送数据；GET方式与POST方式的设置方式不一样
36. }
37. }


41. **function** **ajaxCallback**() {
43. //alert("test");
44. **if**(xmlHttpRequest.readyState == 4) {
46. **if**(xmlHttpRequest.status == 200) {
48. **var** responseText = xmlHttpRequest.responseText;
50. document.getElementById("div1").innerHTML = responseText;
51. }
52. }
53. }
54. </script>

57. </head>
59. <body>
60. <input type=*"button"* value=*"get content from server"* onclick="ajaxSubmit();"> <br/>
61. <input type=*"text"* name=*"value1"* id=*"value1ID"*/><br/>
62. <input type=*"text"* name=*"value2"* id=*"value2ID"*/><br/>
63. <div id=*"div1"*></div>
64. </body>
65. </html>

**相应的服务器端的servlet为AjaxServlet.java:**

**public** **class** AjaxServlet **extends** HttpServlet {

@Override

**protected** **void** doGet(HttpServletRequest req, HttpServletResponse resp)

**throws** ServletException, IOException {

System.*out*.println("doGet invoked!");

process(req, resp);

}

**private** **void** process(HttpServletRequest req, HttpServletResponse resp)

**throws** IOException {

resp.setHeader("pragma", "no-cache");

resp.setHeader("cache-control", "no-cache");

String v1 = req.getParameter("v1");

String v2 = req.getParameter("v2");

String v3 = String.*valueOf*(Integer.*valueOf*(v1) + Integer.*valueOf*(v2));

PrintWriter out = resp.getWriter();

//out.println("Hello World!");

out.println(v3);

out.flush();

}

@Override

**protected** **void** doPost(HttpServletRequest req, HttpServletResponse resp)

**throws** ServletException, IOException {

System.*out*.println("doPost invoked!");

process(req, resp);

}

}

1. **Jquery的口号：write less，do more。**
2. **Jquery的ready()函数与onload属性的区别：**

**onload为window窗口对象的一个属性。指的是当页面加载完了以后且页面中的dom元素的关联关系设定好了以后才去执行onload所关联的那个函数。而jquery中的ready()函数指的是当页面中的dom元素加载完毕后，且页面之间的关联关系还没设置好就立即执行参数中的函数。Window.onload属性只能关联一个函数，后面关联的函数将会覆盖前面关联的函数。而jquery中的ready()函数则能关联多个函数。被关联的函数依次执行例如：**

**<script type="text/javascript">**

**function test1(){**

**alert("Hello");**

**}**

**function test2(){**

**alert("World");**

**}**

window.onload = test1;

//下面这句将覆盖上面那句

window.onload = test2;

**</script>**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**$(document).ready(**

**function() {**

**alert("Hello");**

**}**

**);**

**$(document).ready(**

**function() {**

**alert("World");**

**}**

**);**

**</script>**

**下面为使用原始的javascript代码与使用jquery完成同一功能的代码：**

**<script type="text/javascript">**

**window.onload = function(){**

**var myLinks = document.getElementsByTagName("a");**

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

**myLinks[i].onclick = function(){**

**alert("Hello World");**

**}**

**}**

**}**

**</script>**

**</head>**

**<body>**

**<a href="#">test1</a><br/>**

**<a href="#">test2</a><br/>**

**<a href="#">test3</a><br/>**

**<a href="#">test4</a><br/>**

**<a href="#">test5</a><br/>**

**<a href="#">test6</a><br/>**

**</body>**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**$(document).ready(**

**function(){**

**$("a").click(**

**function(){**

**alert("Hello World");**

**}**

**);**

**}**

**);**

**</script>**

**</head>**

**<body>**

**<a href="#">test1</a><br/>**

**<a href="#">test2</a><br/>**

**<a href="#">test3</a><br/>**

**<a href="#">test4</a><br/>**

**<a href="#">test5</a><br/>**

**<a href="#">test6</a><br/>**

**</body>**

**Jquery中有一个特点：他对基本的javascript对象进行了再封装，转换成了jquery中的对象，并且把javascript中与事件有关的属性如onload、onclick转换成了ready()、click()等方法，这些方法一般都接受一个函数作为参数。**

**Jquery对象与DOM对象的转换方法的实例如下：**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**$(document).ready(**

**function(){**

**var pElement = document.getElementsByTagName("p")[0];**

**//将DOM对象转换为jquery对象**

**var pElementJQuery = $(pElement);**

**//alert("DOM对象的结果：" + pElement.innerHTML);**

**//alert("jQuery对象的结果：" + pElementJQuery.html());**

**var cm = $("#clickMe");//获得jquery对象**

**//将jquery对象转换为DOM对象(第一种方式)**

**var t = cm[0];**

**alert(t.innerHTML);**

**//将jquery对象转换为DOM对象(第二种方式)**

**var s = cm.get(0);**

**alert(s.innerHTML);**

**}**

**);**

**</script>**

**</head>**

**<body>**

**<p id="clickMe">点击我</p>**

**</body>**

**<body>**

**<a id="hello" href="http://www.google.com">Click Me</a>**

**<script type="text/javascript">**

**//alert(document.getElementById("hello").href);**

**if(document.getElementById("hello")){**

**document.getElementById("hello").style.color="red";**

**}**

**</script>**

**</body>**

**<body>**

**<a id="hello" href="http://www.google.com" style="color:red;">Click Me</a>**

**<script type="text/javascript">**

**//$("#hello").css("color","red");写属性**

**alert($("#hello").css("color"));//读属性**

**//alert($("#hello"));即使没有#hello表示的对象，也不为空**

**//alert($("#hello").length);**

**//alert($("#hello")[0]);如果有href属性，默认会获得该属性的值，如果没有该属性，则显示空。而不是null或undefined**

**//alert($("#hello").get(0));同上**

**</script>**

**</body>**

**注意：jquery中，如果函数只包含一个参数，那就是读，包含两个参数就是写。像上面的css()函数一样。注意通过jquery的$("#hello").css("color","red");语句来获取对象的时候，如果并没有#hello的对象，这时jquery并不会报错，而只是忽略掉继续往下进行。这是jquery与传统的javascript不同的地方。这时通过alert($("#hello"))打印出来的仍是一个Object，而并不是传统javascript中的null或者是undefinded。此时通过alert($("#hello").length);语句打印出来的结果是0，而不是undefinded。而alert($("#hello")[0]);和alert($("#hello").get(0));语句也会弹出一个警告框但里面什么也没有，是空的。**

**<body>**

**<table id="table1" width="80%" align="center" border="1">**

**<tbody>**

**<tr><td>Hello</td><td>Hello</td><td>Hello</td><td>Hello</td></tr>**

**<tr><td>world</td><td>world</td><td>world</td><td>world</td></tr>**

**<tr><td>aaaa</td><td>aaaa</td><td>aaaa</td><td>aaaa</td></tr>**

**<tr><td>bbbbb</td><td>bbbbb</td><td>bbbbb</td><td>bbbbb</td></tr>**

**<tr><td>cccc</td><td>cccc</td><td>cccc</td><td>cccc</td></tr>**

**<tr><td>dddd</td><td>dddd</td><td>dddd</td><td>dddd</td></tr>**

**<tr><td>eeee</td><td>eeee</td><td>eeee</td><td>eeee</td></tr>**

**<tr><td>ffff</td><td>ffff</td><td>ffff</td><td>ffff</td></tr>**

**<tr><td>hhhh</td><td>hhhh</td><td>hhhh</td><td>hhhh</td></tr>**

**</tbody>**

**</table>**

**<script type="text/javascript">**

**window.onload = function(){**

**var myTable = document.getElementById("table1");**

**var myTbody = myTable.getElementsByTagName("tbody")[0];**

**var trs = myTbody.getElementsByTagName("tr");**

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

**if(i%2 == 0) {**

**trs[i].style.backgroundColor = "red";**

**} else {**

**trs[i].style.backgroundColor = "green";**

**}**

**}**

**}**

**</script>**

**</body>**

**注意：通过javascript来操纵CSS属性，属性要采取“驼峰”格式，像上面的trs[i].style.backgroundColor一样，其在html中的相应写法为style=”background-color”**

**<body>**

**<input type="checkbox" name="myCheck">**

**<input type="checkbox" name="myCheck" checked="checked">**

**<input type="checkbox" name="myCheck">**

**<input type="checkbox" name="myCheck" checked="checked">**

**<input type="checkbox" name="myCheck">**

**<br/><br/>**

**<input type="button" value="click me" id="myButton">**

**<script type="text/javascript">**

**window.onload = function(){**

**var btn = document.getElementById("myButton");**

**btn.onclick = function(){**

**var count = 0;**

**var myChecks = document.getElementsByName("myCheck");**

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

**if(myChecks[i].checked) {**

**count++;**

**}**

**}**

**alert("count: " + count);**

**}**

**}**

**</script>**

**</body>**

**CSS选择器：**

|  |  |  |  |
| --- | --- | --- | --- |
| 选择器 | 语法 | 描述 | 示例 |
| 标签选择器 | **E{**  **CSS规则**  **}** | **以文档元素作为选择符** | **td{**  **font-size:14px;**  **width:140px;**  **}**  **a{**  **text-docoration:none;**  **}** |
| ID选择器 | **#ID{**  **CSS规则**  **}** | **以文档元素的惟一标识符ID作为选择符** | **#note{**  **font-size:14px;**  **width:140px;**  **}** |
| 类选择器 | **E.className{**  **CSS规则**  **}** | **以文档元素的class作为选择符** | **div.note{**  **font-size:14px;**  **}**  **.dream{**  **font-size:14px;**  **}** |
| 群组选择器 | **E1，E2，E3{**  **CSS规则**  **}** | **多个选择符应用同样的样式规则** | **td,p,div,a {**  **font-size:14px;**  **}** |
| 后代选择器 | **E F{**  **CSS规则**  **}** | **元素E的任意后代元素F(并不一定就是直接的子元素)** | **#links a{**  **Color:red;**  **}** |
| 通配符选择器 | **\*{**  **CSS规则**  **}** | **以文档的所有元素作为选择符** | **\*{**  **Font-size:14px;**  **}** |

**jQuery选择器的分类:**

**--基本选择器(basic)**

**--层次选择器(level)**

**--过滤选择器(filter)**

**--表单选择器(form)**

**1.基本选择器：**

|  |  |  |  |
| --- | --- | --- | --- |
| 选择器 | 描述 | 返回 | 示例 |
| #id | **根据选择的ID匹配一个元素** | **单个元素** | **$(“#test”)选择id为test的元素** |
| .class | **根据给定的类名匹配元素** | **集合元素** | **$(“.test”)选取所有clas为test的元素** |
| Element | **根据给定的元素名匹配所有元素** | **集合元素** | **$(“p”)选取所有的<p>元素** |
| \* | **选取所有的元素** | **集合元素** | **$(“\*”)选取所有的元素** |
| selector1,selector2,  ……,selectorN | **将每一个选择器匹配到的所有元素合并后一起返回** | **集合元素** | **$(“div,span,p.myClass)选取所有<div>,<span>和拥有class为myClass的<p>标签的一组元素** |

**1):ID选择器**

**注意：当采用id选择器进行选择时，如果有多个id值相同的元素，jquery只会选择第一个元素。例如：**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**$("document").ready(**

**function(){**

**//alert($("#test1").get(0).href);**

**//alert($("#test1").html());**

**alert($("#test1").text());**

**}**

**);**

**</script>**

**</head>**

**<body>**

**<a id="test1" href="http://www.google.com"><b>google</b></a>**

**<a id="test1" href="http://www.yahoo.com">baidu</a>**

**</body>**

**警告框弹出的值为**[**http://www.google.com**](http://www.google.com)**，即第一个id为”test1”的<a>元素。它的查找规则为从上倒下查找，如果找到了就不再往下查找。注意上面的html()方法和text()的区别：html()返回调用该方法的jquery元素所对应的DOM元素的开始标签与结束标签之间的所有内容，包括子标签，例如上面的html()的返回值为：<b>google</b>，而text()方法仅仅返回里面的文本内容，不包括子标签。例如上面的text()的返回值为：google。**

**2）：class选择器：**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**$("document").ready(**

**function(){**

**alert($(".test1").length);**

**}**

**);**

**</script>**

**</head>**

**<body>**

**<a class="test1" href="http://www.google.com"><b>google</b></a>**

**<a class="test1" href="http://www.yahoo.com">baidu</a>**

**</body>**

**注意：在使用jquery中的class选择器时，jquery只看html元素中有没有定义class属性，而不看是否真正定义了CSS样式表文件。即只看上文中的<a>元素中是否有class属性，而不看CSS文件中是否真正有名为test1的class选择器。**

**3)元素选择器：**

**<script type="text/javascript">**

**$("document").ready(**

**function(){**

**alert($("a").length);**

**}**

**);**

**</script>**

**</head>**

**<body>**

**<a class="test1" href="http://www.google.com"><b>google</b></a>**

**<a class="test1" href="http://www.yahoo.com">baidu</a>**

**一个基本选择器的例子：**

**<html>**

**<head>**

**<title>jQuery实例</title>**

**<link rel="stylesheet" type="text/css" href="style.css"></link>**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**$("document").ready(**

**function() {**

**$("#button1").click(**

**function(){**

**$("#one").css("background","red");**

**}**

**);**

**$("#button2").click(**

**function(){**

**$(".mini").css("background","green");**

**}**

**);**

**$("#button3").click(**

**function(){**

**$("div").css("background","orange");**

**}**

**);**

**$("#button4").click(**

**function(){**

**$("\*").css("background","blue");**

**}**

**);**

**$("#button5").click(**

**function(){**

**$("span,#two,.mini").css("background","pink");**

**}**

**);**

**}**

**);**

**</script>**

**</head>**

**<body>**

**<h3>基本选择器</h3><br/>**

**<input type="button" value="test1" id="button1">**

**<input type="button" value="test2" id="button2">**

**<input type="button" value="test3" id="button3">**

**<input type="button" value="test4" id="button4">**

**<input type="button" value="test5" id="button5">**

**<br/>**

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

**id为one，class为one的div**

**<div class="mini">class为mini</div>**

**</div>**

**<div class="one" id="two" title="test">**

**class为one，id为two，title为test的div。**

**<div class="mini" title="other">class为mini，title为other</div>**

**<div class="mini" title="test">class为mini，title为test</div>**

**</div>**

**<div class="one">**

**<div class="mini">class为mini</div>**

**<div class="mini">class为mini</div>**

**<div class="mini">class为mini</div>**

**<div class="mini"></div>**

**</div>**

**<div class="one">**

**<div class="mini">class为mini</div>**

**<div class="mini">class为mini</div>**

**<div class="mini">class为mini</div>**

**<div class="mini" title="tesst">class为mini,title为tesst</div>**

**</div>**

**<div class="one" style="display:none">**

**style的display为none的div**

**</div>**

**<div class="hide">class为hide的div</div>**

**<div>**

**包含input的type为hide的div<input type="hidden" size="8"></input>**

**</div>**

**</body>**

**</html>**

**引用的style.css为：**

**div,p,span{**

**width:140px;**

**height:140px;**

**margin:5px;**

**background:#aaa;**

**border:#000 1px solid;**

**font-size:17px;**

**font-family:Verdana;**

**float:left;**

**}**

**div.mini{**

**width:55px;**

**height:55px;**

**background:#aaa;**

**font-size:12px;**

**}**

**div.hide{**

**display:none;**

**}**

**2.层次选择器：**

|  |  |  |  |
| --- | --- | --- | --- |
| 选择器 | 描述 | 返回 | 示例 |
| $(“anscetor descendant”) | **选取anscetor元素里的所有descendant(后代)元素** | **集合元素** | **$(“div span”)选取<div>元素里的所有<span>元素(并不一定是直接的子元素，还包括孙子元素)** |
| $(“parent > child”) | **选取<parent>元素的直接子元素<child>** | **集合元素** | **$(“div > span”)选取<div>元素下的所有直接子元素<span>** |
| $(‘prev + next’) | **选取prev元素后的下一个兄弟元素next** | **集合元素** | **$(‘.one + div’)选取class为one的下一个兄弟元素<div>** |
| $(‘prev ~ siblings) | **选取prev元素后的所有兄弟元素** | **集合元素** | **$(‘.one ~ div’)选取class为one的元素的所有<div>兄弟元素** |
| selector1,selector2,  ……,selectorN | **将每一个选择器匹配到的所有元素合并后一起返回** | **集合元素** | **$(“div,span,p.myClass)选取所有<div>,<span>和拥有class为myClass的<p>标签的一组元素** |

**层次选择器的例子：**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**$("document").ready(**

**function(){**

**$("#button1").click(**

**function(){**

**$('body div').css("background","red");**

**}**

**);**

**$("#button2").click(**

**function(){**

**$('body > div').css("background","blue");**

**}**

**);**

**$("#button3").click(**

**function(){**

**$('.one + div').css("background","green");**

**}**

**);**

**$("#button4").click(**

**function(){**

**$('#two ~ div').css("background","orange");**

**}**

**);**

**}**

**);**

**</script>**

**</head>**

**<body>**

**<h3>基本选择器</h3><br/>**

**<input type="button" value="test1" id="button1">**

**<input type="button" value="test2" id="button2">**

**<input type="button" value="test3" id="button3">**

**<input type="button" value="test4" id="button4">**

**<br/>**

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

**id为one，class为one的div**

**<div class="mini">class为mini</div>**

**</div>**

**<div class="one" id="two" title="test">**

**class为one，id为two，title为test的div。**

**<div class="mini" title="other">class为mini，title为other</div>**

**<div class="mini" title="test">class为mini，title为test</div>**

**</div>**

**<div class="one">**

**<div class="mini">class为mini</div>**

**<div class="mini">class为mini</div>**

**<div class="mini">class为mini</div>**

**<div class="mini"></div>**

**</div>**

**<div class="one">**

**<div class="mini">class为mini</div>**

**<div class="mini">class为mini</div>**

**<div class="mini">class为mini</div>**

**<div class="mini" title="tesst">class为mini,title为tesst</div>**

**</div>**

**<div class="one" style="display:none">**

**style的display为none的div**

**</div>**

**<div class="hide">class为hide的div</div>**

**<div>**

**包含input的type为hide的div<input type="hidden" size="8"></input>**

**</div>**

**</body>**

**层次选择器的等价关系：**

|  |  |  |
| --- | --- | --- |
|  | **选择器** | **方法** |
| **等价关系** | **$(“.one + div”)** | **$(“.one”).next(“div”);** |
|  | **$(“#one ~ div”)** | **$(“#one”).nextAll(“div”);** |

**$(“#one ~ div”)只能选择“#one”后面的同辈<div>元素，而siblings()方法与前后位置无关只要是同辈元素就都能选择。**

**3.过滤选择器：**

1. **基本过滤选择器：**

|  |  |  |  |
| --- | --- | --- | --- |
| 选择器 | 描述 | 返回 | 示例 |
| :first | **返回第一个元素** | **单个元素** | **$(“div:first”)选取所有<div>元素中的第一个<div>元素** |
| :last | **返回最后一个元素** | **单个元素** | **$(“div:last”)选取所有<div>元素中的最后一个<div>元素** |
| :not(selector) | **去除所有与给定选择器匹配的元素** | **集合元素** | **$(“input:not(.myClass)”)选取class不是myClass的所有<input>元素。** |
| :even | **选取索引是偶数的所有元素，索引从0开始** | **集合元素** | **$(“input:even”)选取索引是偶数的所有<input>元素** |
| :odd | **选取索引是奇数的所有元素，索引从0开始** | **集合元素** | **$(“input:odd”)选取索引是奇数的所有<input>元素** |
| :eq(index) | **选取索引等于index的元素** | **单个元素** | **$(“input:eq(1)”)选取索引等于1的<input>元素** |
| :gt(index) | **选取索引大于index的所有元素** | **集合元素** | **$(“input:gt(1)”) 选取索引大于1的<input>元素** |
| :lt(index) | **选取索引小于index的所有元素** | **集合元素** | **$(“input:lt(1)”) 选取索引小于1的<input>元素** |
| :header | **选取所有的标题元素，例如h1,h2,h3,h4,h5，h6等** | **集合元素** | **$(“:header”)选取网页中的所有标题元素** |
| :animated | **选取当前正在执行动画的所有元素** | **集合元素** | **$(“div:animated”)选取正在执行动画的<div>元素。** |
| :contains(“text”) | **选取包含文本内容为text的所有元素** | **集合元素** | **$(“div:contains(‘我’)”)选取包含文本’我’的<div>元素** |
| :empty | **选取不包含子元素或者文本的空元素** | **集合元素** | **$(“div:empty”)选取不包括子元素或者文本的空<div>元素** |
| :has(selector) | **选取含有选择器所匹配的元素的元素** | **集合元素** | **$(“div:has(p)”)选取包含<p>元素的<div>元素** |
| :parent | **选取含有子元素或者文本的元素** | **集合元素** | **$(“div:parent”)选取含有子元素(包含文本)的<div>元素** |

**内容过滤选择器的例子：**

**<script type="text/javascript">**

**$(function(){**

**$("#button1").click(**

**function(){**

**$("div:contains('test')").css("background","red");**

**});**

**$("#button2").click(**

**function(){**

**$("div:empty").css("background","green");**

**}**

**);**

**$("#button3").click(**

**function(){**

**$("div:has('.mini')").css("background","blue");**

**}**

**);**

**$("#button4").click(**

**function(){**

**$("div:parent").css("background","#abccff");**

**}**

**);**

**});**

**</script>**

**可见性过滤选择器：**

|  |  |  |  |
| --- | --- | --- | --- |
| 选择器 | 描述 | 返回 | 示例 |
| :hidden | **选取所有不可见的元素** | **集合元素** | **$(“:hidden”);选取所有不可见的元素，包括<input type=”hidden”/>;和<div style=”display:none;”/>;和**  **<div style=”visibility:hidden;”/>等元素。如果只想选取<input>元素，使用$(“input:hidden”)** |
| :visible | **选取所有可见的元素** | **集合元素** | **$(“div:visible”)选取所有可见的<div>元素。** |

**可见性过滤选择器的例子：**

**<script type="text/javascript">**

**$().ready(**

**function(){**

**$("#button1").click(**

**function(){**

**//alert($("div:hidden").length);**

**//alert($("input:hidden").length);**

**$("div:hidden").show(5000).css("background","blue");**

**}**

**);**

**$("#button2").click(**

**function(){**

**$("div:visible").css("background","red");**

**}**

**);**

**}**

**);**

**</script>**

**属性过滤选择器：**

|  |  |  |  |
| --- | --- | --- | --- |
| 选择器 | 描述 | 返回 | 示例 |
| [attribute] | **选取拥有此属性的元素** | **集合元素** | **$(“div[id]”)选取拥有属性id的<div>元素** |
| [attribute=value] | **选取属性的值为value的所有元素** | **集合元素** | **$(“div[title=test]”)选取title等于”test”的<div>元素** |
| [attribute != value] | **选取属性的值不等于value的所有元素** | **集合元素** | **$(“div[title != test]”)选取title不等于”test”的<div>元素（包括没有title属性的<div>元素）** |
| [attribute ^= value] | **选取属性的值以value开头的所有元素** | **集合元素** | **$(“div[title ^= test]”)选取title属性以”test”开头的<div>元素** |
| [attribute $= value] | **选取属性的值以value结尾的所有元素** | **集合元素** | **$(“div[title $= test]”)选取title属性以”test”结尾的<div>元素** |
| [attribute \*= value] | **选取属性的值包含有value的所有元素** | **集合元素** | **$(“div[title \*= test]”)选取title属性的值包含有”test”的<div>元素** |
| [selector1][selector2][selectorN] | **用属性选择器合并成一个复合属性选择器，满足多个条件。没选择一次，缩小一次范围** | **集合元素** | **$(div[id][title$=’test’])选取拥有属性id并且title属性值以’test’结尾的<div>元素** |

**属性过滤选择器的例子：**

**<script type="text/javascript">**

**$(function(){**

**$("#button1").click(**

**function(){**

**$("div[title]").css("background","green");**

**}**

**);**

**$("#button2").click(**

**function(){**

**$("div[title=test]").css("background","red");**

**}**

**);**

**$("#button3").click(**

**function(){**

**$("div[title != test]").css("background","pink");**

**}**

**);**

**$("#button4").click(**

**function(){**

**$("div[title ^= test]").css("background","pink");**

**}**

**);**

**$("#button5").click(**

**function(){**

**$("div[title $= st]").css("background","pink");**

**}**

**);**

**$("#button6").click(**

**function(){**

**$("div[title \*= st]").css("background","pink");**

**}**

**);**

**$("#button7").click(**

**function(){**

**$("div[id][title ^= t][title $= t]").css("background","pink");**

**}**

**);**

**});**

**</script>**

**子元素过滤选择器的例子：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 选择器 | 描述 | 返回 | 示例 | |
| :nth-child  (index/even/odd/  equation) | **选取每个父元素下的第index个元素或者奇偶元素(index从1开始算起)** | **集合元素** | | **:eq(index)只匹配一个元素，而nth-child将为每一个父元素匹配子元素，并且nth-child(indx)的index是从1开始的，而eq(index)的index是从0开始的** |
| :first-child | **选取每个父元素的第一个子元素** | **集合元素** | | **:first只返回单个元素，而:first-child选择符将为每个父元素选择第一个子元素例如：**  **$(“ul li:first-child”);**  **选取每个<ul>中的第一个<li>元素** |
| :last-child | **选取每个父元素的最后一个子元素** | **集合元素** | | **:last只返回单个元素，**  **:last-child会为每个父元素选择最后一个子元素。**  **$(“ul li:last-child”);**  **选取每个<ul>中的最后一个<li>元素** |
| :only-child | **如果父元素中只含有惟一的子元素则会被匹配，如果父元素中还含有其它的子元素则不会被匹配** | **集合元素** | | **$(“ul li:only-child”)**  **在<ul>中选取是惟一的子元素的**  **<li>元素** |

**子元素过滤选择器的例子：**

**<script type="text/javascript">**

**$(function(){**

**$("#button1").click(**

**function(){**

**$("div.one :nth-child(2)").css("background","red");**

**}**

**);**

**$("#button2").click(**

**function(){**

**$("div.one :first-child").css("background","green");**

**}**

**);**

**$("#button3").click(**

**function(){**

**$("div.one :last-child").css("background","pink");**

**}**

**);**

**$("#button4").click(**

**function(){**

**$("div.one :only-child").css("background","orange");**

**}**

**);**

**});**

**</script>**

**表单对象属性过滤选择器：**

|  |  |  |  |
| --- | --- | --- | --- |
| 选择器 | 描述 | 返回 | 示例 |
| :enabled | **选取所有可用元素** | **集合元素** | **$(“#form1 :enabled”)**  **选取id为form1的表单内的所有可用元素** |
| :disabled | **选取所有不可用元素** | **集合元素** | **$(“#form2 :disabled”)**  **选取id为form2的表单内的所有不可用元素** |
| :checked | **选取所有被选中的元素(单选框、复选框)** | **集合元素** | **$(“input:checked”);**  **选取所有被选中的<input>元素** |
| :selected | **选取所有被选中的选项元素(下拉列表)** | **集合元素** | **$(“select:selected”)**  **选取所有被选中的选项元素** |

**表单对象属性过滤器的例子：**

**<script type="text/javascript">**

**$(function(){**

**$("#button1").click(**

**function(){**

**$("#form1 input:gt(1):enabled").val("Hello World");**

**}**

**);**

**$("#button2").click(**

**function(){**

**$("#form1 input:disabled").val("welcome");**

**}**

**);**

**$(":checkbox").click(**

**function(){**

**$("div:first").html("<font color='red'><b>" + $("input:checked").length + "</b></font>");**

**}**

**);**

**$("select").change(**

**function(){**

**var str = "";**

**$("select :selected").each(**

**function(){**

**str += $(this).text() + ",";**

**}**

**);**

**$("div:last").html("<b>" + str + "</b>");**

**}**

**);**

**});**

**</script>**

**</head>**

**<body>**

**<h3>表单对象属性过滤选择器</h3><br/>**

**<form id="form1" action="#">**

**<!--**

**//当使用被注释的语句时，<form>元素中应加 onsubmit=return false;**

**<button id="button1">对表单内可用的input赋值操作</button>**

**<button id="button2">对表单内不可用的input赋值操作</button>**

**-->**

**<input type="button" id="button1" value="对表单内可用的input赋值操作"/>**

**<input type="button" id="button2" value="对表单内不可用的input赋值操作"/>**

**<br/><br/>**

**可用元素：<input name="add" value="可用文本框"/> <br/>**

**不可用元素：<input name="email" disabled="disabled" value="不可用文本框"/><br/>**

**可用元素：<input name="che" value="可用文本框"/> <br/>**

**不可用元素：<input name="name" disabled="disabled" value="不可用文本框"/><br/>**

**<br/>**

**多选框：<br/>**

**<input type="checkbox" name="newsletter" checked="checked" value="test1"/>test1**

**<input type="checkbox" name="newsletter" value="test2"/>test2**

**<input type="checkbox" name="newsletter" value="test3"/>test3**

**<input type="checkbox" name="newsletter" checked="checked" value="test4"/>test4**

**<input type="checkbox" name="newsletter" value="test5"/>test5**

**<br/><br/>**

**下拉列表1:<br/>**

**<select name="test" multiple="multiple" style="height:100px;">**

**<option>上海</option>**

**<option selected="selected">香港</option>**

**<option>北京</option>**

**<option selected="selected">河北</option>**

**<option>广东</option>**

**<option>重庆</option>**

**</select>**

**<br/><br/>**

**下拉列表2：<br/>**

**<select name="test2">**

**<option>海南</option>**

**<option>上海</option>**

**<option selected="selected">北京</option>**

**<option>云南</option>**

**<option>广州</option>**

**<option>贵州</option>**

**</select>**

**<br/><br/>**

**<div></div>**

**<div></div>**

**</form>**

**表单选择器：**

|  |  |  |  |
| --- | --- | --- | --- |
| 选择器 | 描述 | 返回 | 示例 |
| :input | **选取所有的<input>、<textarea>、**  **<select>和<button>函数。** | **集合元素** | **$(“:input”) 选取所有的<input>、<textarea>、**  **<select>和<button>函数。** |
| :text | **选取所有的单行文本框** | **集合元素** | **$(“:text”) 选取所有的单行文本框** |
| :password | **选取所有的密码框** | **集合元素** | **$(“:password”) 选取所有的密码框** |
| :radio | **选取所有的单选按钮** | **集合元素** | **$(“:radion”)选取所有的单选按钮** |
| :checkbox | **选取所有的复选框** | **集合元素** | **$(“:checkbox”)选取所有的复选框** |
| :submit | **选取所有的提交按钮** | **集合元素** | **$(“:submit”) 选取所有的递交按钮** |
| :image | **选取所有的图像按钮** | **集合元素** | **$(“:image”) 选取所有的图像按钮** |
| :reset | **选取所有的重置按钮** | **集合元素** | **$(“:reset”) 选取所有的重置按钮** |
| :button | **选取所有的按钮** | **集合元素** | **$(“:button”) 选取所有的按钮** |
| :file | **选取所有的文件上传域** | **集合元素** | **$(“:file”) 选取所有的文件上传域** |
| :hidden | **选取所有的不可见元素** | **集合元素** | **$(“:hidden”) 选取所有的不可见元素** |

**表单选择器的例子：**

**<script type="text/javascript">**

**$(function(){**

**//alert($("#form1 :text").length);**

**//alert($("#form1 :password").length);**

**//alert($("#form1 :radio").length);**

**//alert($("#form1 :checkbox").length);**

**//alert($("#form1 :submit").length);**

**//alert($("#form1 :input").length);**

**alert($("#form1 input").length);**

**});**

**</script>**

**</head>**

**<body>**

**<form id="form1" action="#">**

**<input type="button" value="Button"><br/>**

**<input type="checkbox" name="c">1<input type="checkbox" name="c">2<input type="checkbox" name="c">3<br/>**

**<input type="file"/><br/>**

**<input type="hidden"/><div style="display:none">test</div><br/>**

**<input type="password"/><br/>**

**<input type="radio" name="a">1<input type="radio" name="a">2<br/>**

**<input type="reset" /><br/>**

**<input type="submit" value="提交" /><br/>**

**<input type="text"><br/>**

**<select><option>Option</option></select><br/>**

**<textarea rows="5" cols="20"></textarea><br/>**

**<button>Button</button><br/>**

**</form>**

**注意$("#form1 :input")与$("#form1 input")的区别：**

**$("#form1 :input")选取form1中所有的<input>、<textarea>、<select>和<button>的元素。它属于表单选择器的范畴。$("#form1 input")表示选取form1中所有的<input>元素。它属于层次选择器的范畴。**

**下面为实现表格按行交错换颜色的例子：**

**<script type="text/javascript">**

**/\***

**$().ready(**

**function(){**

**$("#table1 tr:odd").css("background","red");**

**$("#table1 tr:even").css("background","blue");**

**}**

**);**

**$(function(){**

**$("#table1 tr:odd").css("background","red");**

**});**

**$(function(){**

**$("#table1 tr:even").css("background","green");**

**});**

**\*/**

**$(function(){**

**$("#table1 tr:odd").css("background","green");**

**$("#table1 tr:even").css("background","blue");**

**});**

**</script>**

**script type="text/javascript">**

**$(**

**function(){**

**$("#myButton").click(**

**function(){**

**alert('被选中的复选框为：' + $("input:checked").length);**

**}**

**);**

**}**

**);**

**</script>**

**<script type="text/javascript">**

**$(**

**function(){**

**//选择class为test的元素当中的隐藏子元素（子代选择器）**

**alert($(".test :hidden").length);**

**//选取隐藏的class为test的元素**

**alert($(".test:hidden").length);**

**}**

**);**

**</script>**

**</head>**

**<body>**

**<div class="test">**

**<div style="display:none">aaaa</div>**

**<div style="display:none">bbbb</div>**

**<div style="display:none">cccc</div>**

**<div class="test" style="display:none">dddd</div>**

**</div>**

**<div class="test" style="display:none">eeee</div>**

**<div class="test" style="display:none">ffff</div>**

**</body>**

**Jquery操纵DOM：**

1. **htmlDOM的常见操作：**

**1).查找节点**

**2).插入节点**

**3).删除节点**

**4).复制节点**

**5).替换节点**

**6).包裹节点**

**<script type="text/javascript">**

**$(**

**function(){**

**var p = $("p");**

**var li = $("ul li:eq(1)");**

**var title = p.attr('title');**

**var title2 = li.attr("title");**

**var text = li.text();**

**alert(title);**

**alert(title2);**

**alert(text);**

**li.attr("title","myTitle");**

**}**

**);**

**</script>**

**Jquery中操纵对象的方法：**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| append() | **向每个匹配的元素内部添加内容** | **HTML代码：**  **<p>我想说：</p>**  **jQuery代码：**  **$(“p”).append(“<b>你好</b>”);**  **结果：**  **<p>我想说：<b>你好</b></p>** |
| appendTo() | **将所有匹配的元素追加到指定的元素中。实际上，使用该方法是颠倒了常规的$(“A”).append(“B”)方法。即不是将B追加到A中，而是将A追加到B中。** | **HTML代码：**  **<p>我想说：</p>**  **jQuery代码：**  **$(“<b>你好</b>”).appendTo(“p”);**  **结果：**  **<p>我想说：<b>你好</b></p>** |
| prepend() | **向每个匹配的元素内部前置内容** | **HTML代码：**  **<p>我想说：</p>**  **jQuery代码：**  **$(“p”).prepend(“<b>你好</b>”);**  **结果：**  **<p><b>你好</b>我想说： </p>** |
| prependTo() | **将所有匹配的元素前置到指定的元素中。实际上，使用该方法是颠倒了常规的$(“A”).prepend(“B”)方法。即不是将B前置到A中，而是将A前置到B中。** | **HTML代码：**  **<p>我想说：</p>**  **jQuery代码：**  **$(“<b>你好</b>”)**  **.prependTo(“p”);**  **结果：**  **<p><b>你好</b>我想说： </p>** |
| after() | **在每个匹配的元素之后插入内容** | **HTML代码：**  **<p>我想说：</p>**  **jQuery代码：**  **$(“p”).after(“<b>你好</b>”);**  **结果：**  **<p>我想说：</p><b>你好</b>** |
| insertAfter() | **将所有匹配的元素插入到指定元素的后面。实际上，使用该方法是颠倒了常规的(“A”).insertAfter(“B”)**  **方法。即不是将B插入到A后面，而是将A插入到B后面。** | **HTML代码：**  **<p>我想说：</p>**  **jQuery代码：**  **$(“<b>你好</b>”)**  **.insertAfter(“p”);**  **结果：**  **<p>我想说：</p><b>你好</b>** |
| before() | **在每个匹配的元素之前插入内容** | **HTML代码：**  **<p>我想说：</p>**  **jQuery代码：**  **$(“p”)**  **.before(“<b>你好</b>”);**  **结果：**  **<b>你好</b><p>我想说：</p>** |
| insertBefore() | **将所有匹配的元素插入到指定元素的前面。实际上，使用该方法是颠倒了常规的$(“A”).insertBefore(“B”)方法。即不是将B插入到A前面，而是将A插入到B前面。** | **HTML代码：**  **<p>我想说：</p>**  **jQuery代码：**  **$(“<b>你好</b>”)**  **.insertBefore(“p”);**  **结果：**  **<b>你好</b>**  **<p>我想说：</p>** |

**1).添加元素的例子：**

**<script type="text/javascript">**

**var addItems =function(){**

**/\***

**document.getElementById("div1").innerHTML = "";**

**var value = parseInt(document.getElementById("itemsNumber").value);**

**for(var i=0; i<value; i++) {**

**var Input = document.createElement("input");**

**Input.setAttribute("type", "text");**

**var br = document.createElement("br");**

**//console.log(Input);**

**//注意：只有容器元素中才能追加子元素**

**document.getElementById("div1").appendChild(Input);**

**document.getElementById("div1").appendChild(br);**

**}**

**\*/**

**//使用jquery实现同样的功能**

**document.getElementById("div1").innerHTML = "";**

**var value = parseInt($("#itemsNumber").val());**

**for(var i=0; i<value; i++) {**

**$("#div1").append("<input type='text'><br/>");**

**}**

**}**

**</script>**

**</head>**

**<body>**

**<input type="text" id="itemsNumber"/>**

**<input type="button" id="btn" onclick="addItems();" value="add Items"/><br/>**

**<div id="div1"></div>**

**</body>**

**2).插入元素的实例：**

**<meta http-equiv="Content-Type" content="text/html;charset=GB2312"/>**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**$(function(){**

**/\***

**$("ul").append("<li title='abc'>Hello</li>")**

**.append("<li title='xyz'>World</li>");**

**\*/**

**$("<li title='abc'>Hello</li>").appendTo("ul");**

**$("<li title='abc'>World</li>").appendTo("ul");**

**});**

**</script>**

**</head>**

**<body>**

**<p title="Hello World">Cacey:你认为我对你怎么样呢？</p>**

**<ul>**

**<li title="1">好</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**</body>**

**<script type="text/javascript">**

**$(function(){**

**var li1 = $("<li title='hello'>hello</li>");**

**var li2 = $("<li title='world'>world</li>");**

**var li3 = $("<li title='welcome'>welcome</li>");**

**$("ul").prepend(li1);**

**$("ul").append(li3);**

**$("ul li:eq(4)").after(li2);**

**});**

**</script>**

**</head>**

**<body>**

**<p title="Hello World">Cacey:你认为我对你怎么样呢？</p>**

**<ul>**

**<li title="1">好</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**</body>**

**<script type="text/javascript">**

**$(function(){**

**//实现节点的移动**

**$("ul li:eq(2)").insertAfter("ul li:eq(4)");**

**});**

**</script>**

**</head>**

**<body>**

**<p title="Hello World">Cacey:你认为我对你怎么样呢？</p>**

**<ul>**

**<li title="1">好</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**</body>**

**3).移除节点的实例：**

**<script type="text/javascript">**

**$(function(){**

**//remove()方法会返回被移除的节点的jquery对象**

**var removedLi = $("ul li:eq(3)").remove();**

**removedLi.appendTo($("ul"));**

**//根据条件删除**

**$("ul li").remove("li[title != 2]");**

**//清空元素的文本内容**

**$("ul li:eq(3)").empty();**

**});**

**</script>**

**</head>**

**<body>**

**<p title="Hello World">Cacey:你认为我对你怎么样呢？</p>**

**<ul>**

**<li title="1">好</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**</body>**

**//动态添加与删除文件域 的例子**

**<html>**

**<head>**

**<title>jQuery实例</title>**

**<meta http-equiv="Content-Type" content="text/html;charset=GB2312"/>**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**/\***

**//javascript实现方式**

**var addMore = function(){**

**var div1 = document.getElementById("div1");**

**var file1 = document.createElement("input");**

**var delButton = document.createElement("input");**

**var br = document.createElement("br");**

**file1.setAttribute("type","file");**

**delButton.setAttribute("type","button");**

**delButton.setAttribute("value","Remove");**

**delButton.onclick = function(){**

**div1.removeChild(file1);**

**div1.removeChild(delButton);**

**div1.removeChild(br);**

**}**

**div1.appendChild(file1);**

**div1.appendChild(delButton);**

**div1.appendChild(br);**

**}**

**\*/**

**var count = 0;**

**$(**

**function(){**

**$("input[type=button]").click(**

**function(){**

**var br = $("<br/>");**

**var Input = $("<input type='file'/>");**

**var button = $("<input type='button' value='Remove" + (++count) + "'/>");**

**$("#div1").append(br).append(Input).append(button);**

**button.click(**

**function(){**

**br.remove();**

**Input.remove();**

**button.remove();**

**}**

**);**

**}**

**);**

**}**

**);**

**</script>**

**</head>**

**<body>**

**<input type="file"/><input type="button" id ="btn" value="more..."/><br/>**

**<div id="div1"></div>**

**</body>**

**</html>**

**4).jquery中对象的克隆：**

**<script type="text/javascript">**

**$(function(){**

**$("ul li").click(**

**function(){**

**//clone()方法传入参数true表示克隆出的对象与被克隆的对象具有同样的属性，默认为true，反之为false**

**$(this).clone(true).appendTo('ul');**

**}**

**);**

**});**

**</script>**

**元素的包裹：**

**<script type="text/javascript">**

**$(function(){**

**//$("p").wrap("<a href='http://www.google.com'></a>");**

**$("p").wrapInner("<a href='http://www.google.com'></a>");**

**});**

**</script>**

**与属性操作有关的方法：**

**<script type="text/javascript">**

**$(function(){**

**$("input:eq(0)").click(**

**function(){**

**//$("p").attr("title","welcome");**

**$("p").attr({"title":"welcome", "height":"17px","width":"260px"});**

**}**

**);**

**$("input:eq(1)").click(**

**function(){**

**alert($("p").attr("title"));**

**}**

**);**

**$("input:eq(2)").click(**

**function(){**

**$("p").removeAttr("height");**

**}**

**);**

**});**

**</script>**

**与样式有关的操作：**

**<style type="text/css">**

**.high{**

**font-weight:bold;**

**color:red;**

**}**

**.another{**

**font-style:italic;**

**color:green;**

**}**

**</style>**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("input:eq(0)").click(**

**function(){**

**alert($("p").attr("class"));**

**}**

**);**

**$("input:eq(1)").click(**

**function(){**

**$("p").attr("class","high");**

**}**

**);**

**$("input:eq(2)").click(**

**function(){**

**$("p").addClass("high");**

**}**

**);**

**$("input:eq(3)").click(**

**function(){**

**$("p").removeClass("high");**

**}**

**);**

**$("input:eq(4)").click(**

**function(){**

**//把class的属性设为空值，即移除所有的class属性**

**$("p").removeClass();**

**}**

**);**

**$("input:eq(5)").click(**

**function(){**

**//如果有another属性，则把它移除，如果没有，则添加。即在两种属性之间切换**

**$("p").toggleClass("another");**

**}**

**);**

**$("input:eq(6)").click(**

**function(){**

**//alert($("p").hasClass("another"));**

**//is()用户判断元素是否是一个选择器，这只是它的其中一个用途**

**//alert($("p").is(".another"));**

**alert($("p").is(":animated"));**

**}**

**);**

**});**

**</script>**

**注意：方法addClass()用于追加样式,如果已经有样式，它不会替换原有的样式，而是取并集。attr()可用于设置样式属性和其它属性，如果已有class属性，则会替换掉原有class属性，而不是取并集。**

**<style type="text/css">**

**.high{**

**font-weight:bold;**

**color:red;**

**}**

**.another{**

**font-style:italic;**

**color:green;**

**}**

**</style>**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("input:eq(0)").click(**

**function(){**

**alert($("p").html());**

**}**

**);**

**$("input:eq(1)").click(**

**function(){**

**alert($("p").text());**

**}**

**);**

**$("input:eq(2)").click(**

**function(){**

**//不进行转义，直接把表现结果显示在界面上**

**$("p").html("<a href='http://www.google.com'>google</a>");**

**}**

**);**

**$("input:eq(3)").click(**

**function(){**

**//对于html标签会进行转义，从而在界面上显示文本内容**

**$("p").text("<a href='http://www.google.com'>google</a>");**

**}**

**);**

**});**

**</script>**

**</head>**

**<body>**

**<input type="button" value="button1">**

**<input type="button" value="button2">**

**<input type="button" value="button3">**

**<input type="button" value="button4">**

**<input type="button" value="button5">**

**<input type="button" value="button6">**

**<input type="button" value="button7">**

**<br/>**

**<p title="Hello World" class="hello"><b>Cacey:你认为我对你怎么样呢？</b></p>**

**<ul>**

**<li title="1">好</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**</body>**

**<script type="text/javascript">**

**$(function(){**

**$("input:eq(0)").click(**

**function(){**

**alert($("p").html());**

**}**

**);**

**$("input:eq(1)").click(**

**function(){**

**alert($("p").text());**

**}**

**);**

**$("input:eq(2)").click(**

**function(){**

**//不进行转义，直接把表现结果显示在界面上**

**$("p").html("<a href='http://www.google.com'>google</a>");**

**}**

**);**

**$("input:eq(3)").click(**

**function(){**

**//对于html标签'<'符号会转义成&lt;，而对于'>'会转义成&gt;，从而在界面上显示文本内容本身**

**$("p").text("<a href='http://www.google.com'>google</a>");**

**}**

**);**

**$("input:eq(4)").click(**

**function(){**

**//打印第五个按钮上显示的文本值**

**alert($(this).val());**

**}**

**);**

**$("input:eq(5)").click(**

**function(){**

**//修改第六个按钮上显示的文本值**

**$(this).val("hello");**

**}**

**);**

**});**

**</script>**

**</head>**

**<body>**

**<input type="button" value="button1">**

**<input type="button" value="button2">**

**<input type="button" value="button3">**

**<input type="button" value="button4">**

**<input type="button" value="button5">**

**<input type="button" value="button6">**

**<br/>**

**<p title="Hello World" class="hello"><b>Cacey:你认为我对你怎么样呢？</b></p>**

**<ul>**

**<li title="1">好</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**</body>**

**<script type="text/javascript">**

**$(function(){**

**$('#username').focus(**

**function(){**

**var value = $(this).val();**

**//if(value == 'username'){**

**//利用DOM对象的fefaultValue属性**

**if(value == this.defaultValue){**

**$(this).val('');**

**}**

**}**

**);**

**$('#username').blur(**

**function(){**

**var value = $(this).val();**

**if(value == '') {**

**//$(this).val('username');**

**$(this).val(this.defaultValue);**

**}**

**}**

**);**

**});**

**</script>**

**</head>**

**<body>**

**<input type=“text" id="username" value="username"/><br/>**

**<input type="password" id="password" value="password"/><br/>**

**<input type="button" value="click me"/>**

**</body>**

**使用jQuery对元素进行遍历：**

**<script type="text/javascript">**

**$(function(){**

**var v1 = $("body").children();**

**var v2 = $("p").children();**

**var v3 = $("ul").children();**

**alert(v1.length);**

**alert(v2.length);**

**alert(v3.length);**

**for(var i=0; i<v3.length; i++) {**

**alert(v3[i].innerHTML);**

**}**

**});**

**</script>**

**</head>**

**<body>**

**<p title="Hello World" class="hello">Caocao:你认为我对你怎么样呢？</p>**

**<ul>**

**<li title="1">好</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**</body>**

**<script type="text/javascript">**

**$(function(){**

**var v1 = $("p").next();**

**//alert(v1.html());**

**var v2 = $("p").prev();**

**//alert(v2.html());**

**var v3 = $("p").siblings();**

**alert(v3.length);**

**});**

**</script>**

**</head>**

**<body>**

**<ul>**

**<li title="1">aaa</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**<ul>**

**<li title="1">bb</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**<ul>**

**<li title="1">好</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**<ul>**

**<li title="1">好</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**<ul>**

**<li title="1">好</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**<p title="Hello World" class="hello">Caocao:你认为我对你怎么样呢？</p>**

**<ul>**

**<li title="1">好</li>**

**<li title="2">很好</li>**

**<li title="3">非常好</li>**

**<li title="4">特别好</li>**

**<li title="5">太好了</li>**

**<li title="6">好的无法描述了</li>**

**</ul>**

**</body>**

**Jquery中对事件的处理：**

**事件冒泡的例子：**

**<script type="text/javascript">**

**var method1 = function(){**

**alert('div clicked');**

**}**

**var method2 = function(){**

**alert('span clicked');**

**}**

**var method3 = function(){**

**alert('body clicked');**

**}**

**</script>**

**<style type="text/css">**

**\*{margin:0; padding:0}**

**body{font-size:13px;line-height:130%;padding:60px}**

**#content{width:220px;border:1px solid #0050D0;background:#96e555;}**

**span{width:200px; margin:10px; background:#666666; cursor:pointer; color:white; display:block;}**

**p{width:200px; background:#888; color:white; height:16px;}**

**</style>**

**</head>**

**<body onclick="method3();">**

**<div id="content" onclick="method1();">**

**外层div元素**

**<span onclick="method2();">内层span元素</span>**

**外层div元素**

**</div>**

**<div id="msg">**

**</div>**

**</body>**

**如何阻止事件冒泡的例子：**

**<style type="text/css">**

**\*{margin:0; padding:0}**

**body{font-size:13px;line-height:130%;padding:60px}**

**#content{width:220px;border:1px solid #0050D0;background:#96e555;}**

**span{width:200px; margin:10px; background:#666666; cursor:pointer; color:white; display:block;}**

**p{width:200px; background:#888; color:white; height:16px;}**

**</style>**

**<script type="text/javascript" src="jquery-1.4.4.js"></script>**

**<script type="text/javascript">**

**var method1 = function(){**

**alert('div clicked');**

**}**

**var method2 = function(event){**

**alert('span clicked');**

**event.stopPropagation();**

**//下面的方法用来解决浏览器处理事件的兼容性问题**

**event.preventDefault();**

**}**

**var method3 = function(){**

**alert('body clicked');**

**}**

**</script>**

**</head>**

**<body onclick="method3();">**

**<div id="content" onclick="method1();">**

**外层div元素**

**<span onclick="method2(event);">内层span元素</span>**

**外层div元素**

**</div>**

**<div id="msg">**

**</div>**

**</body>**

**jQuery中事件的冒泡：**

**<style type="text/css">**

**\*{margin:0; padding:0}**

**body{font-size:13px;line-height:130%;padding:60px}**

**#content{width:220px;border:1px solid #0050D0;background:#96e555;}**

**span{width:200px; margin:10px; background:#666666; cursor:pointer; color:white; display:block;}**

**p{width:200px; background:#888; color:white; height:16px;}**

**</style>**

**<script type="text/javascript" src="jquery-1.4.4.js"></script>**

**<script type="text/javascript">**

**$(**

**function(){**

**//为span元素绑定click事件**

**$('span').bind('click',function(){**

**var txt = $('#msg').html() + "<p>内层span元素被点击.</p>";**

**$('#msg').html(txt);**

**});**

**//为div元素绑定click事件**

**$("#content").bind('click',function(){**

**var txt = $('#msg').html() + "<p>外层div元素被点击.</p>";**

**$('#msg').html(txt);**

**});**

**//为body元素绑定click事件**

**$('body').bind('click',function(){**

**var txt = $('#msg').html() + "<p>body元素被点击.</p>";**

**$('#msg').html(txt);**

**});**

**}**

**);**

**</script>**

**</head>**

**<body>**

**<div id="content">**

**外层div元素**

**<span>内层span元素</span>**

**外层div元素**

**</div>**

**<div id="msg">**

**</div>**

**</body>**

**Jquery中阻止事件的冒泡：**

**<style type="text/css">**

**\*{margin:0; padding:0}**

**body{font-size:13px;line-height:130%;padding:60px}**

**#content{width:220px;border:1px solid #0050D0;background:#96e555;}**

**span{width:200px; margin:10px; background:#666666; cursor:pointer; color:white; display:block;}**

**p{width:200px; background:#888; color:white; height:16px;}**

**</style>**

**<script type="text/javascript" src="jquery-1.5.js"></script>**

**<script type="text/javascript">**

**$(**

**function(){**

**//为span元素绑定click事件**

**$('span').bind('click',function(event){**

**var txt = $('#msg').html() + "<p>内层span元素被点击.</p>";**

**$('#msg').html(txt);**

**//阻止事件的冒泡**

**return false;**

**});**

**//为div元素绑定click事件**

**$("#content").bind('click',function(){**

**var txt = $('#msg').html() + "<p>外层div元素被点击.</p>";**

**$('#msg').html(txt);**

**//阻止事件的冒泡**

**return false;**

**});**

**//为body元素绑定click事件**

**$('body').bind('click',function(){**

**var txt = $('#msg').html() + "<p>body元素被点击.</p>";**

**$('#msg').html(txt);**

**});**

**}**

**);**

**</script>**

**</head>**

**<body>**

**<div id="content">**

**外层div元素**

**<span>内层span元素</span>**

**外层div元素**

**</div>**

**<div id="msg">**

**</div>**

**</body>**

**Jquery与javascript的加载时间比较：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title></title>**

**<script src="jquery-1.5.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**var startTime = new Date().getTime();**

**$(document).ready(function(){**

**test1();**

**})**

**function test1(){**

**var endTime2 = new Date().getTime();**

**var a = endTime2 - startTime;**

**$("<div>jQuery的ready() : "+a+" ms</div>").appendTo("body");**

**}**

**function test2(){**

**var endTime1 = new Date().getTime();**

**var b = endTime1 - startTime;**

**$("<p>JavaScript的window.onload : "+b+" ms</p>").appendTo("body");**

**}**

**</script>**

**</head>**

**<body onload="test2();">**

**<img src="http://www.google.com.hk/logos/2011/cezanne11-hp.jpg" style="width:200px;height:200px;"/>**

**</body>**

**</html>**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=GB2312" />**

**<title>4-2-2</title>**

**<script src="jquery-1.5.js" type="text/javascript"></script>**

**<link rel="stylesheet" type="text/css" href="../../css/style.css" />**

**<script type="text/javascript">**

**$(function(){**

**$("#panel h5.head").bind("click",function(){**

**$(this).next().show(10000);//show()方法中可以传入slow、normal、fast等参数**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<div id="panel">**

**<h5 class="head">什么是jQuery?</h5>**

**<div class="content">**

**jQuery是继Prototype之后又一个优秀的JavaScript库，它是一个由 John Resig 创建于2006年1月的开源项目。jQuery凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发人员遍历HTML文档、操作DOM、处理事件、执行动画和开发Ajax。它独特而又优雅的代码风格改变了JavaScript程序员的设计思路和编写程序的方式。**

**</div>**

**</div>**

**</body>**

**</html>**

**鼠标滑过的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>4-2-4</title>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<link rel="stylesheet" type="text/css" href="../../css/style.css" />**

**<script type="text/javascript">**

**$(function(){**

**$("#panel h5.head").bind("mouseover",function(){**

**$(this).next().show(2000);**

**});**

**$("#panel h5.head").bind("mouseout",function(){**

**$(this).next().hide(2000);**

**})**

**})**

**</script>**

**</head>**

**<body>**

**<div id="panel">**

**<h5 class="head">什么是jQuery?</h5>**

**<div class="content">**

**jQuery是继Prototype之后又一个优秀的JavaScript库，它是一个由 John Resig 创建于2006年1月的开源项目。jQuery凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发人员遍历HTML文档、操作DOM、处理事件、执行动画和开发Ajax。它独特而又优雅的代码风格改变了JavaScript程序员的设计思路和编写程序的方式。**

**</div>**

**</div>**

**</body>**

**</html>**

**Click、mouseOver、mouseOut等事件的简写：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>4-2-5</title>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<link rel="stylesheet" type="text/css" href="../../css/style.css" />**

**<script type="text/javascript">**

**$(function(){**

**$("#panel h5.head").mouseover(function(){**

**$(this).next().show();**

**});**

**$("#panel h5.head").mouseout(function(){**

**$(this).next().hide();**

**})**

**})**

**</script>**

**</head>**

**<body>**

**<div id="panel">**

**<h5 class="head">什么是jQuery?</h5>**

**<div class="content">**

**jQuery是继Prototype之后又一个优秀的JavaScript库，它是一个由 John Resig 创建于2006年1月的开源项目。jQuery凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发人员遍历HTML文档、操作DOM、处理事件、执行动画和开发Ajax。它独特而又优雅的代码风格改变了JavaScript程序员的设计思路和编写程序的方式。**

**</div>**

**</div>**

**</body>**

**</html>**

**鼠标悬停的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>4-3-1</title>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<link rel="stylesheet" type="text/css" href="../../css/style.css" />**

**<script type="text/javascript">**

**$(function(){**

**$("#panel h5.head").hover(function(){**

**$(this).next().show();**

**},function(){**

**$(this).next().hide();**

**})**

**})**

**</script>**

**</head>**

**<body>**

**<div id="panel">**

**<h5 class="head">什么是jQuery?</h5>**

**<div class="content">**

**jQuery是继Prototype之后又一个优秀的JavaScript库，它是一个由 John Resig 创建于2006年1月的开源项目。jQuery凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发人员遍历HTML文档、操作DOM、处理事件、执行动画和开发Ajax。它独特而又优雅的代码风格改变了JavaScript程序员的设计思路和编写程序的方式。**

**</div>**

**</div>**

**</body>**

**</html>**

**toggle事件的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>4-3-2</title>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<link rel="stylesheet" type="text/css" href="../../css/style.css" />**

**<script type="text/javascript">**

**$(function(){**

**$("#panel h5.head").toggle(function(){**

**$(this).next().show();**

**},function(){**

**$(this).next().hide();**

**})**

**})**

**</script>**

**</head>**

**<body>**

**<div id="panel">**

**<h5 class="head">什么是jQuery?</h5>**

**<div class="content">**

**jQuery是继Prototype之后又一个优秀的JavaScript库，它是一个由 John Resig 创建于2006年1月的开源项目。jQuery凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发人员遍历HTML文档、操作DOM、处理事件、执行动画和开发Ajax。它独特而又优雅的代码风格改变了JavaScript程序员的设计思路和编写程序的方式。**

**</div>**

**</div>**

**</body>**

**</html>**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>4-3-3</title>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<link rel="stylesheet" type="text/css" href="../../css/style.css" />**

**<script type="text/javascript">**

**$(function(){**

**$("#panel h5.head").toggle(function(){**

**$(this).next().toggle();**

**},function(){**

**$(this).next().toggle();**

**})**

**})**

**/\*$(function(){**

**$("#panel h5.head").click(function(){**

**$(this).next().toggle();**

**})**

**})\*/**

**</script>**

**</head>**

**<body>**

**<div id="panel">**

**<h5 class="head">什么是jQuery?</h5>**

**<div class="content">**

**jQuery是继Prototype之后又一个优秀的JavaScript库，它是一个由 John Resig 创建于2006年1月的开源项目。jQuery凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发人员遍历HTML文档、操作DOM、处理事件、执行动画和开发Ajax。它独特而又优雅的代码风格改变了JavaScript程序员的设计思路和编写程序的方式。**

**</div>**

**</div>**

**</body>**

**</html>**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>4-3-4</title>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<link rel="stylesheet" type="text/css" href="../../css/style.css" />**

**<style type="text/css">**

**.highlight{ background: orange; }**

**</style>**

**<script type="text/javascript">**

**$(function(){**

**$("#panel h5.head").toggle(function(){**

**$(this).addClass("highlight");**

**$(this).next().show();**

**},function(){**

**$(this).removeClass("highlight");**

**$(this).next().hide();**

**});**

**})**

**</script>**

**</head>**

**<body>**

**<div id="panel">**

**<h5 class="head">什么是jQuery?</h5>**

**<div class="content">**

**jQuery是继Prototype之后又一个优秀的JavaScript库，它是一个由 John Resig 创建于2006年1月的开源项目。jQuery凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发人员遍历HTML文档、操作DOM、处理事件、执行动画和开发Ajax。它独特而又优雅的代码风格改变了JavaScript程序员的设计思路和编写程序的方式。**

**</div>**

**</div>**

**</body>**

**</html>**

**阻止表单提交的默认行为：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title></title>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("#sub").bind("click",function(event){**

**var username = $("#username").val(); //获取元素的值**

**if(username==""){ //判断值是否为空**

**$("#msg").html("<p>文本框的值不能为空.</p>"); //提示信息**

**event.preventDefault(); //阻止默认行为 ( 表单提交 )**

**}**

**})**

**})**

**</script>**

**</head>**

**<body>**

**<form action="test.html">**

**用户名：<input type="text" id="username" />**

**<br/>**

**<input type="submit" value="提交" id="sub"/>**

**</form>**

**<div id="msg"></div>**

**</body>**

**</html>**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title></title>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("#sub").bind("click",function(event){**

**var username = $("#username").val(); //获取元素的值**

**if(username==""){ //判断值是否为空**

**$("#msg").html("<p>文本框的值不能为空.</p>"); //提示信息**

**//return false有两个作用：1.防止表单重复提交，2.防止事件冒泡传播**

**return false;**

**}**

**})**

**})**

**</script>**

**</head>**

**<body>**

**<form action="test.html">**

**用户名：<input type="text" id="username" />**

**<br/>**

**<input type="submit" value="提交" id="sub"/>**

**</form>**

**<div id="msg"></div>**

**</body>**

**</html>**

**事件对象的一些属性：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<title></title>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<!-- 引入jQuery -->**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script>**

**$(function(){**

**$("a").click(function(event) {**

**alert(event.type);//获取事件类型**

**return false;//阻止链接跳转**

**//event.preventDefault();**

**//event.stopPropagation();**

**});**

**})**

**</script>**

**</head>**

**<body>**

**<a href='http://google.com'>click me .</a>**

**</body>**

**</html>**

**获取事件源对象的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<title></title>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<!-- 引入jQuery -->**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script>**

**$(function(){**

**$("a[href=http://google.com]").click(function(event) {**

**alert(event.target.href);//获取触发事件的<a>元素的href属性值**

**return false;//阻止链接跳转**

**});**

**})**

**</script>**

**</head>**

**<body>**

**<a href='http://google.com'>click me .</a>**

**</body>**

**</html>**

**事件的连续绑定：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>4-6-1</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { font-size: 13px; line-height: 130%; padding: 60px; }**

**p {width:200px;background:#888;color:white;height:16px;}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$('#btn').bind("click", function(){**

**$('#test').append("<p>我的绑定函数1</p>");**

**}).bind("click", function(){**

**$('#test').append("<p>我的绑定函数2</p>");**

**}).bind("click", function(){**

**$('#test').append("<p>我的绑定函数3</p>");**

**});**

**})**

**</script>**

**</head>**

**<body>**

**<button id="btn">点击我</button>**

**<div id="test"></div>**

**</body>**

**</html>**

**取消绑定的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>4-6-2</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { font-size: 13px; line-height: 130%; padding: 60px; }**

**p {width:200px;background:#888;color:white;height:16px;}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$('#btn').bind("click", function(){**

**$('#test').append("<p>我的绑定函数1</p>");**

**}).bind("click", function(){**

**$('#test').append("<p>我的绑定函数2</p>");**

**}).bind("click", function(){**

**$('#test').append("<p>我的绑定函数3</p>");**

**});**

**$('#delAll').click(function(){**

**$('#btn').unbind(); //$("#btn").unbind();**

**});**

**})**

**</script>**

**</head>**

**<body>**

**<button id="btn">点击我</button>**

**<div id="test"></div>**

**<button id="delAll">删除所有事件</button>**

**</body>**

**</html>**

**取消某个特定的绑定：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>Panel</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { font-size: 13px; line-height: 130%; padding: 60px; }**

**p {width:200px;background:#888;color:white;height:16px;}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$('#btn').bind("click", myFun1 = function(){**

**$('#test').append("<p>我的绑定函数1</p>");**

**}).bind("click", myFun2 = function(){**

**$('#test').append("<p>我的绑定函数2</p>");**

**}).bind("click", myFun3 = function(){**

**$('#test').append("<p>我的绑定函数3</p>");**

**});**

**$('#delTwo').click(function(){**

**$('#btn').unbind("click",myFun2);**

**});**

**})**

**</script>**

**</head>**

**<body>**

**<button id="btn">点击我</button>**

**<div id="test"></div>**

**<button id="delTwo">删除第二个事件</button>**

**</body>**

**</html>**

**事件只被绑定一次的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>4-6-4</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { font-size: 13px; line-height: 130%; padding: 60px; }**

**p {width:200px;background:#888;color:white;height:16px;}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$('#btn').one("click", function(){**

**$('#test').append("<p>我的绑定函数1</p>");**

**}).one("click", function(){**

**$('#test').append("<p>我的绑定函数2</p>");**

**}).one("click", function(){**

**$('#test').append("<p>我的绑定函数3</p>");**

**});**

**})**

**</script>**

**</head>**

**<body>**

**<button id="btn">点击我</button>**

**<div id="test"></div>**

**</body>**

**</html>**

**模拟用户的某个操作的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title></title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { font-size: 13px; line-height: 130%; padding: 60px; }**

**p {width:200px;background:#888;color:white;height:16px;}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$('#btn').bind("click", function(){**

**$('#test').append("<p>我的绑定函数1</p>");**

**}).bind("click", function(){**

**$('#test').append("<p>我的绑定函数2</p>");**

**}).bind("click", function(){**

**$('#test').append("<p>我的绑定函数3</p>");**

**});**

**//$('#btn').trigger("click");//触发用户的click事件**

**$('#btn').click(); //同样的效果**

**})**

**</script>**

**</head>**

**<body>**

**<button id="btn">点击我</button>**

**<div id="test"></div>**

**</body>**

**</html>**

**模拟自定义事件的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title></title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { font-size: 13px; line-height: 130%; padding: 60px; }**

**p {width:200px;background:#888;color:white;height:16px;}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$('#btn').bind("myClick", function(){**

**$('#test').append("<p>我的自定义事件.</p>");**

**});**

**$('#btn').click(function(){**

**$(this).trigger("myClick");**

**}).trigger("myClick");**

**})**

**</script>**

**</head>**

**<body>**

**<button id="btn">点击我</button>**

**<div id="test"></div>**

**</body>**

**</html>**

**模拟带参数的自定义事件的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title></title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { font-size: 13px; line-height: 130%; padding: 60px; }**

**p {width:200px;background:#888;color:white;height:16px;}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$('#btn').bind("myClick", function(event, message1, message2){**

**$('#test').append( "<p>"+message1 + message2 +"</p>");**

**});**

**$('#btn').click(function(){**

**$(this).trigger("myClick",["我的自定义","事件"]);**

**}).trigger("myClick",["hello","world"]);//通过数组来传递自定义参数的值**

**})**

**</script>**

**</head>**

**<body>**

**<button id="btn">点击我</button>**

**<div id="test"></div>**

**</body>**

**</html>**

**同时绑定多个事件的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<title></title>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<style>**

**.over{**

**color:red;**

**background:blue;**

**}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("div").bind("mouseover mouseout", function(){**

**$(this).toggleClass("over");**

**});**

**})**

**</script>**

**</head>**

**<body>**

**<div style="width:100px;height:50px;">滑入.</div>**

**</body>**

**</html>**

**使用jquery实现动画渐变的效果的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>Panel</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { font-size: 13px; line-height: 130%; padding: 60px }**

**#panel { width: 300px; border: 1px solid #0050D0 }**

**.head { padding: 5px; background: #96E555; cursor: pointer }**

**.content { padding: 10px; text-indent: 2em; border-top: 1px solid #0050D0;display:block; }**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("#panel h5.head").toggle(function(){**

**$(this).next().fadeOut(3000);//$(this).next().fadeOut(4000);**

**},function(){**

**$(this).next().fadeIn(3000);**

**})**

**})**

**</script>**

**</head>**

**<body>**

**<div id="panel">**

**<h5 class="head">什么是jQuery?</h5>**

**<div class="content">**

**jQuery是继Prototype之后又一个优秀的JavaScript库，它是一个由 John Resig 创建于2006年1月的开源项目。jQuery凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发人员遍历HTML文档、操作DOM、处理事件、执行动画和开发Ajax。它独特而又优雅的代码风格改变了JavaScript程序员的设计思路和编写程序的方式。**

**</div>**

**</div>**

**</body>**

**</html>**

**Jquery实现从上倒下收缩和从下到上展开：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>Panel</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { font-size: 13px; line-height: 130%; padding: 60px }**

**#panel { width: 300px; border: 1px solid #0050D0 }**

**.head { padding: 5px; background: #96E555; cursor: pointer }**

**.content { padding: 10px; text-indent: 2em; border-top: 1px solid #0050D0;display:block; }**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("#panel h5.head").toggle(function(){**

**$(this).next().slideUp(3000);**

**},function(){**

**$(this).next().slideDown(3000);**

**})**

**})**

**</script>**

**</head>**

**<body>**

**<div id="panel">**

**<h5 class="head">什么是jQuery?</h5>**

**<div class="content">**

**jQuery是继Prototype之后又一个优秀的JavaScript库，它是一个由 John Resig 创建于2006年1月的开源项目。jQuery凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发人员遍历HTML文档、操作DOM、处理事件、执行动画和开发Ajax。它独特而又优雅的代码风格改变了JavaScript程序员的设计思路和编写程序的方式。**

**</div>**

**</div>**

**</body>**

**</html>**

**实现从左向右移动的效果：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>Panel</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { padding: 60px }**

**#panel {position: relative; width: 100px; height:100px;border: 1px solid #0050D0;background: #96E555; cursor: pointer}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("#panel").click(function(){**

**//表示每次单击的时候都在原有的基础上距左边500px像素，每次移动所费的时间为//3秒，移动完成时弹出一个警告框**

**$(this).animate({left: "+=500px"}, 3000,function(){alert(3);});**

**})**

**})**

**</script>**

**</head>**

**<body>**

**<div id="panel"></div>**

**</body>**

**</html>**

**同时进行的多重动画的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>Panel</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { padding: 60px }**

**#panel {position: relative; width: 100px; height:100px;border: 1px solid #0050D0;background: #96E555; cursor: pointer}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("#panel").click(function(){**

**//移动的同时增加高度**

**$(this).animate({left: "500px",height:"200px"}, 3000);**

**})**

**})**

**</script>**

**</head>**

**<body>**

**<div id="panel"></div>**

**</body>**

**</html>**

**不同时进行的多重动画的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>Panel</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { padding: 60px }**

**#panel {position: relative; width: 100px; height:100px;border: 1px solid #0050D0;background: #96E555; cursor: pointer}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**//先移动后增高**

**$("#panel").click(function(){**

**$(this).animate({left: "500px"}, 3000)**

**.animate({height: "200px"}, 3000);**

**})**

**})**

**</script>**

**</head>**

**<body>**

**<div id="panel"></div>**

**</body>**

**</html>**

**设置透明度的动画的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>Panel</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { padding: 60px }**

**#panel {position: relative; width: 100px; height:100px;border: 1px solid #0050D0;background: #96E555; cursor: pointer}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("#panel").css("opacity", "0.5");//ʨ׃²»͸ķ¶ƍ**

**$("#panel").click(function(){**

**$(this).animate({left: "400px", height:"200px" ,opacity: "1"}, 3000)**

**.animate({top: "200px" , width :"200px"}, 3000 )**

**.fadeOut("slow");**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<div id="panel"></div>**

**</body>**

**</html>**

**实现未排队动画的例子：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>Panel</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { padding: 60px }**

**#panel {position: relative; width: 100px; height:100px;border: 1px solid #0050D0;background: #96E555; cursor: pointer}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("#panel").css("opacity", "0.5");//设置不透明度**

**$("#panel").click(function(){**

**$(this).animate({left: "400px", height:"200px" ,opacity: "1"}, 3000)**

**.animate({top: "200px" , width :"200px"}, 3000 )**

**.css("border","5px solid blue");**

**});**

**});**

**//注意：边线的效果与第一个动画的效果是同时进行的**

**</script>**

**</head>**

**<body>**

**<div id="panel"></div>**

**</body>**

**</html>**

**排队实现动画效果：**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

**<html xmlns="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<title>Panel</title>**

**<style type="text/css">**

**\*{margin:0;padding:0;}**

**body { padding: 60px }**

**#panel {position: relative; width: 100px; height:100px;border: 1px solid #0050D0;background: #96E555; cursor: pointer}**

**</style>**

**<script src="../../scripts/jquery-1.3.1.js" type="text/javascript"></script>**

**<script type="text/javascript">**

**$(function(){**

**$("#panel").css("opacity", "0.5");//设置不透明度**

**$("#panel").click(function(){**

**$(this).animate({left: "400px", height:"200px" ,opacity: "1"}, 3000)**

**.animate({top: "200px" , width :"200px"}, 3000 ,function(){**

**$(this).css("border","5px solid blue");**

**})**

**//注意：边线的变动是同动画的效果不是同时进行的 ，通过回调函数来实现**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<div id="panel"></div>**

**</body>**

**</html>**

**json**

**json是javascript object notation的缩写。Json与xml相比他传输的数据量较小，是基于文本的，javascript的一个子集。它具有轻量级与易于解析的特点。**

**注意：json的MIME TYPE是application/json，而不是我们常见的text/html。**

**Json就是Ajax中的‘x’，即用来取代xml进行数据传输的。我们可以通过XmlHttpRequest来获取responseText，然后再把responseText解析成json对象，方法如下：**

**responseData = eval(‘(‘ + responseText + ‘)’);或者采取如下第二种方法：**

**responseData = responseText.parseJSON();**

**json没有命名空间的概念、json是不可扩展的、json不需要验证器、json不是xml。**

**Json的官方网站的地址为：http：//www.json.org**

**如果需要在项目中使用json库，则可以在源代码目录下建一个命名为org.json的包，然后把我们下载的所有源代码复制到该包中即可。注意因为json中使用了junit测试框架，因此我们也要导入junit测试包，不然会报错。或者把json包中的Test.java文件删除也可。一个json的例子：**

**package org.daixy.json;**

**import org.json.JSONArray;**

**import org.json.JSONException;**

**import org.json.JSONObject;**

**public class Test {**

**public static void main(String[] args) throws Exception {**

**String jsonContent = "{'Hello' : 'world', 'xyz' : 'abc'}";**

**JSONObject jsonObject = new JSONObject(jsonContent);**

**String str1 = jsonObject.getString("Hello");**

**String str2 = jsonObject.getString("xyz");**

**System.out.println(str1);**

**System.out.println(str2);**

**System.out.println("---------------------------------------");**

**jsonContent = "[{'hello' : 333, 'abc' : false, 'xyz' : {‘a’ : 1, ‘b’ : ‘ab’}}, {'hello' : 555, 'abc':true, 'xyz' : {‘a’ : ‘2’, ‘b’ : ‘ba’}}]";**

**JSONArray jsonArray = new JSONArray(jsonContent);**

**for(int i=0; i<jsonArray.length(); i++) {**

**JSONObject jsonObject2 = jsonArray.getJSONObject(i);**

**int value1 = jsonObject2.getInt("hello");**

**boolean value2 = jsonObject2.getBoolean("abc");**

JSONObject jsonObject3 = jsonObject2.getJSONObject("xyz");

**int** value3 = jsonObject3.getInt("a");

String value4 = jsonObject3.getString("b");

**System.out.println(value1);**

**System.out.println(value2);**

**System.out.println(value3);**

**}**

**}**

**}**

**另一个有用的json库为google的gson库。一个Gson的例子如下：**

**package** org.daixy.json;

**import** java.util.ArrayList;

**import** java.util.List;

**import** com.google.gson.Gson;

**public** **class** Test2 {

**public** **static** **void** main(String[] args) {

Person person = **new** Person();

person.setUsername("zhangsan");

person.setPassword("123456");

person.setAddress("beijing");

person.setAge(30);

List<String> list = **new** ArrayList<String>();

list.add("hello");

list.add("world");

list.add("Hello World");

person.setList(list);

Gson gson = **new** Gson();

//通过Gson把javabean对象转换成json对象

String result = gson.toJson(person);

System.*out*.println(result);

}

}

**注意：Gson把list属性的名字作为生成的json对象的属性的名字，值作为生成对象的属性的值。根据复合属性的类型不同而生成不同类型的值。**

**在web项目中整合jquery库：**

**新建一个web项目，然后在WebRoot目录下新建一个scripts文件夹，把下载的jquery文件放入该目录下即可。如果要使用jquery脚本库中的功能，在jsp文件中使用<script>**

**标签将其导入即可。**

**在web页面中采用json格式来发送异步请求的例子如下：**

<%@ page language=*"java"* import=*"java.util.\*"* pageEncoding=*"UTF-8"*%>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<base href=*"*<%=basePath%>*"*>

<title>My JSP 'index.jsp' starting page</title>

<meta http-equiv=*"pragma"* content=*"no-cache"*>

<meta http-equiv=*"cache-control"* content=*"no-cache"*>

<meta http-equiv=*"expires"* content=*"0"*>

<meta http-equiv=*"keywords"* content=*"keyword1,keyword2,keyword3"*>

<meta http-equiv=*"description"* content=*"This is my page"*>

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

<script type=*"text/javascript"* src=*"scripts/jquery-1.5.js"*></script>

<script type=*"text/javascript"*>

$(

**function**(){

$("#button1").click(

**function**(){

$.ajax(

{//以json格式发送请求

//type:"POST",

type:"GET",

url:"MyServlet",

//dataType可以省略，默认为html

dataType:"html",

//data用来存放提交请求的参数值

data:{"param1":$("#param1").val(), "param2":$("#param2").val()},

success:**function**(returnedData){

$("#result").val(returnedData);

}

}

);

}

);

}

);

</script>

</head>

<body>

<input type=*"text"* id=*"param1"*/> +

<input type=*"text"* id=*"param2"*/> =

<input type=*"text"* id=*"result"*/>

<input type=*"button"* value=*" get content from server "* id=*"button1"*/>

</body>

</html>

**一个返回xml格式响应数据ajax的例子：**

**客户端代码如下：**

<%@ page language=*"java"* import=*"java.util.\*"* pageEncoding=*"UTF-8"*%>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<base href=*"*<%=basePath%>*"*>

<title>My JSP 'xml.jsp' starting page</title>

<meta http-equiv=*"pragma"* content=*"no-cache"*>

<meta http-equiv=*"cache-control"* content=*"no-cache"*>

<meta http-equiv=*"expires"* content=*"0"*>

<meta http-equiv=*"keywords"* content=*"keyword1,keyword2,keyword3"*>

<meta http-equiv=*"description"* content=*"This is my page"*>

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

<script type=*"text/javascript"* src=*"scripts/jquery-1.5.js"*></script>

<script type=*"text/javascript"*>

$(**function**(){

/\*

$("#button1").click(function(){

$.ajax({

type:"POST",

url:"XmlServlet",

dataType:"xml",

data:{"name":$("#name").val()},

success:function(returnedData){

//$(returnedData)是一个xml对象，因此可以调用其find()来找到其函数

var id = $(returnedData).find("id").text();

//alert(id);

var name = $(returnedData).find("name").text();

var age = $(returnedData).find("age").text();

var address = $(returnedData).find("address").text();

var html = "<table width='60%' border='1' align='center'>"

+ "<tr><th>id</th><th>name</th><th>age</th><th>address</th></tr>"

+"<tr align='center'><td>" + id +"</td><td>" + name + "</td><td>" + age + "</td><td>"

+ address + "</td></tr>"

+ "</table>";

//清空界面中上次显示的表格

$("#theBody table:eq(0)").remove();

$("#theBody").append(html);

}

});

});

\*/

/\*

$("#button1").click(function(){

//使用$.ajax()方法的简化版$.post()方法来提交请求

$.post("XmlServlet",

{//json格式的请求参数

"name":$("#name").val()

},function(returnedData,status){

//成功返回时执行的回调函数

//$(returnedData)是一个xml对象，因此可以调用其find()来找到其函数

var id = $(returnedData).find("id").text();

//alert(id);

var name = $(returnedData).find("name").text();

var age = $(returnedData).find("age").text();

var address = $(returnedData).find("address").text();

var html = "<table width='60%' border='1' align='center'>"

+ "<tr><th>id</th><th>name</th><th>age</th><th>address</th></tr>"

+"<tr align='center'><td>" + id +"</td><td>" + name + "</td><td>" + age + "</td><td>"

+ address + "</td></tr>"

+ "</table>";

//清空界面中上次显示的表格

$("#theBody table:eq(0)").remove();

$("#theBody").append(html);

});

});

\*/

$("#button1").click(**function**(){

//使用$.ajax()方法的简化版$.get()方法来提交请求

$.get("XmlServlet",

{//json格式的请求参数

"name":$("#name").val()

},**function**(returnedData,status){

//成功返回时执行的回调函数

//$(returnedData)是一个xml对象，因此可以调用其find()来找到其函数

**var** id = $(returnedData).find("id").text();

//alert(id);

**var** name = $(returnedData).find("name").text();

**var** age = $(returnedData).find("age").text();

**var** address = $(returnedData).find("address").text();

**var** html = "<table width='60%' border='1' align='center'>"

+ "<tr><th>id</th><th>name</th><th>age</th><th>address</th></tr>"

+"<tr align='center'><td>" + id +"</td><td>" + name + "</td><td>" + age + "</td><td>"

+ address + "</td></tr>"

+ "</table>";

//清空界面中上次显示的表格

$("#theBody table:eq(0)").remove();

$("#theBody").append(html);

});

});

});

</script>

</head>

<body id=*"theBody"*>

<select id=*"name"*>

<option value=*"zhangsan"*>zhangsan</option>

<option value=*"lisi"*>lisi</option>

</select>

<input type=*"button"* id=*"button1"* value=*" get content from server "*/>

</body>

</html>

**服务器端代码如下：**

**package** com.daixy.ajax;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** org.dom4j.Document;

**import** org.dom4j.DocumentHelper;

**import** org.dom4j.Element;

**import** org.dom4j.io.OutputFormat;

**import** org.dom4j.io.XMLWriter;

**import** com.daixy.model.Person;

**public** **class** XmlServlet **extends** HttpServlet {

@Override

**protected** **void** doPost(HttpServletRequest req, HttpServletResponse resp)

**throws** ServletException, IOException {

String name = req.getParameter("name");

Person person = **new** Person();

//根据name来生成数据，相当于根据name条件来从后台查询数据

**if**("zhangsan".equals(name)) {

person.setId(1);

person.setName("zhangsan");

person.setAddress("beijing");

person.setAge(30);

}**else** {

person.setId(2);

person.setName("lisi");

person.setAddress("shanghai");

person.setAge(20);

}

//使用dom4j或jdom来构建xml对象

Document document = DocumentHelper.*createDocument*();

Element rootElement = document.addElement("users");

rootElement.addComment("This is a comment!");

Element userElement = rootElement.addElement("user");

Element idElement = userElement.addElement("id");

Element nameElement = userElement.addElement("name");

Element ageElement = userElement.addElement("age");

Element addressElement = userElement.addElement("address");

idElement.setText(person.getId() + "");

nameElement.setText(person.getName());

ageElement.setText(person.getAge() + "");

addressElement.setText(person.getAddress());

//告诉客户端服务器返回的数据类型、以及禁用缓存

resp.setContentType("text/xml; charset=utf-8");

resp.setHeader("pragma", "no-cache");

resp.setHeader("cache-control", "no-cache");

PrintWriter out = resp.getWriter();

//设置返回文本的输出格式，这里为了方便我们查看，实际开发时并不需要设定

OutputFormat format = OutputFormat.*createPrettyPrint*();

format.setEncoding("utf-8");

XMLWriter xmlWriter = **new** XMLWriter(out,format);

xmlWriter.write(document);

out.flush();

}

@Override

**protected** **void** doGet(HttpServletRequest req, HttpServletResponse resp)

**throws** ServletException, IOException {

**this**.doPost(req, resp);

}

}

**一个采用json格式返回响应数据的例子**

**客户端代码：**

<%@ page language=*"java"* import=*"java.util.\*"* pageEncoding=*"UTF-8"*%>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<base href=*"*<%=basePath%>*"*>

<title>My JSP 'json.jsp' starting page</title>

<meta http-equiv=*"pragma"* content=*"no-cache"*>

<meta http-equiv=*"cache-control"* content=*"no-cache"*>

<meta http-equiv=*"expires"* content=*"0"*>

<meta http-equiv=*"keywords"* content=*"keyword1,keyword2,keyword3"*>

<meta http-equiv=*"description"* content=*"This is my page"*>

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

<script type=*"text/javascript"* src=*"scripts/jquery-1.5.js"*></script>

<script type=*"text/javascript"*>

$(**function**(){

$("#button1").click(**function**(){

$.get("GsonServlet",{},**function**(returnedData, status){

**var** html = "<table width='60%' border='1' align='center'>"

+ "<tr><th>id</th><th>name</th><th>companyAddress</th><th>homeAddress</th></tr>";

//alert(returnedData.length);

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

**var** id = returnedData[i].id;

**var** name = returnedData[i].name;

**var** companyAddress = returnedData[i].address.companyAddress;

**var** homeAddress = returnedData[i].address.homeAddress;

html += "<tr align='center'><td>" + id + "</td><td>" + name + "</td><td>" + companyAddress

+ "</td><td>" + homeAddress + "</td></tr>";

}

html += "</table>";

$("#theBody table:eq(0)").remove();

$("#theBody").append(html);

});

});

});

</script>

</head>

<body id=*"theBody"*>

<input type=*"button"* value=*"get json content from server"* id=*"button1"*/>

</body>

</html>

**服务器端代码：**

**package** com.daixy.servlet;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.util.ArrayList;

**import** java.util.List;

**import** com.daixy.model.Address;

**import** com.daixy.model.People;

**import** com.google.gson.Gson;

**import** javax.servlet.ServletException;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**public** **class** GsonServlet **extends** HttpServlet {

@Override

**protected** **void** doGet(HttpServletRequest req, HttpServletResponse resp)

**throws** ServletException, IOException {

List<People> list = **new** ArrayList<People>();

People people1 = **new** People();

Address address1 = **new** Address();

address1.setHomeAddress("beijing");

address1.setCompanyAddress("shanghai");

people1.setId(1);

people1.setName("zhangsan");

people1.setAddress(address1);

People people11 = **new** People();

people11.setId(5);

people11.setName("zhangsan11");

People people12 = **new** People();

people12.setId(6);

people12.setName("zhangsan12");

List<People> list1 = **new** ArrayList<People>();

list1.add(people11);

list1.add(people12);

people1.setFrieds(list1);

System.*out*.println("------------------------");

People people2 = **new** People();

Address address2 = **new** Address();

address2.setCompanyAddress("jiling");

address2.setHomeAddress("liaoling");

people2.setId(2);

people2.setName("lisi");

people2.setAddress(address2);

List<People> list2 = **new** ArrayList<People>();

People people21 = **new** People();

people21.setId(3);

people21.setName("lisi21");

People people22 = **new** People();

people22.setId(4);

people22.setName("lisi22");

list2.add(people21);

list2.add(people22);

people2.setFrieds(list2);

list.add(people1);

list.add(people2);

//通过Gson框架把复杂的数据转换成json格式的数据

Gson gson = **new** Gson();

String result = gson.toJson(list);

//向客户端发送josn格式的响应数据

resp.setContentType("application/json; charset=utf-8");

resp.setHeader("pragma", "no-cache");

resp.setHeader("cache-control", "no-cache");

PrintWriter out = resp.getWriter();

out.println(result);

out.flush();

}

}

**一个利用google进行图片搜索的例子：**

**package com.daixy.image.google;**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStream;**

**import java.io.InputStreamReader;**

**import java.net.MalformedURLException;**

**import java.net.URL;**

**import java.net.URLConnection;**

**import java.util.ArrayList;**

**import java.util.List;**

**public class Test {**

**public static void main(String[] args) throws Exception {**

**List<String> list = new ArrayList<String>();**

**for(int i=0; i<8; i++) {**

**String str = "https://ajax.googleapis.com/ajax/services/search/images?key=ABQIAAAA3CV1bg6DmftnvoD2FfyM5RSo6\_Y-lPd\_7a5rWdJRsISKKsAXgBSon1WwaU0OZl7fWNi5oz5B6zaYNQ&q=macbook&rsz=large&v=1.0&";**

**str += "start=" + 8 \* i;**

**URLConnection conn = new URL(str).openConnection();**

**InputStream is = conn.getInputStream();**

**InputStreamReader isr = new InputStreamReader(is);**

**BufferedReader br = new BufferedReader(isr);**

**StringBuffer buffer = new StringBuffer();**

**String line = null;**

**while(null != (line = br.readLine())){**

**buffer.append(line);**

**}**

**list.add(buffer.toString());**

**br.close();**

**isr.close();**

**is.close();**

**System.out.println(list.size());**

**for(String s : list) {**

**System.out.println(s);**

**}**

**}**

**}**

**}**