# JQuery 基础

## 一、JQuery 简介

JQuery 是js 的一个脚本框架，实际上就是对js的一个再次封装，使得在实现网页的动态处理时，比原生js要更简单，加快前端的开发效率，用起来要比js更灵活。

## 二、JQuery 实际应用

因为JQuery 是第三方公司在原生js 的基础上研发的。所以需要第三方，JQuery js源码包的支持。在实际的开发工程中，我们需要先下载好JQuery 包，在需要JQuery 脚本的网页中，引用JQuery包即可。

|  |
| --- |
| <script type="text/javascript" src="js/jquery.min.js"></script> |

## 三、选择器

### 1、ID选择器

语法：$("id属性值")

作用：根据元素的id属性值，匹配元素节点。

范例：

|  |
| --- |
| <body > <p id="p">这是一个段落</p> </body> <script>  $("#p").css("color","red");  $("#p").css("font-size","25px"); </script> |

### 2、CLASS选择器

语法：$("class属性值")

作用：根据元素的class属性值，匹配元素节点。

范例：

|  |
| --- |
| <body > <p class="pc1">这是一个段落</p> <p >这是一个段落</p> </body>  <script>  $(".pc1").css("color","red"); </script> |

### 3、元素选择器

语法：$("元素名")

作用：根据元素的名称，匹配所有元素节点。

范例：

|  |
| --- |
| <body > <p id="p1" class="pc1">这是一个段落</p> <p >这是一个段落</p> </body> <script>  $("p").css("color","red"); </script> |

### 4、通配符选择器

语法：$("\*")

作用：匹配所有元素节点。

范例：

|  |
| --- |
| <body > <p id="p1" class="pc1">这是一个段落</p> <p >这是一个段落</p> <b>这是一个段落</b> <i>这是一个段落</i> </body> <script>  $("\*").css("color","red"); </script> |

### 5、并集选择器

语法：$("select1,select2,select3...")

作用：一次性匹配多种类型的节点元素。

范例：

|  |
| --- |
| <body > <p id="p1" class="pc1">这是一个段落</p> <p >这是一个段落</p> <b>这是一个段落</b> <i>这是一个段落</i> </body> <script>  $("b,i").css("color","red"); </script> |

### 6、后代选择器

语法：$("select1 select2 ")

作用：匹配select1元素节点中，所有的select2的后代元素。

范例：

|  |
| --- |
| <body > <p id="p1" class="pc1">这是一个段落</p> <p >这是一个段落  <b>这个加粗</b>  你好我好大家好  <b>这个加粗</b> </p> <b>这是一个段落</b> </body> <script>  $("body b").css("color","red"); </script> |

### 7、子元素选择器

语法：$("select1 > select2 ")

作用：匹配select1元素节点中，所有名称为select2的子元素。

范例：

|  |
| --- |
| <body > <p id="p1" class="pc1">这是一个段落</p> <p >这是一个段落  <b>这个加粗</b>  你好我好大家好  <b>这个加粗</b> </p> <b>这是一个段落</b> </body> <script>  $("body>b").css("color","red"); </script> |

### 8、相邻元素选择器

语法：$("select1 + select2 ")

作用：匹配与select1元素节点，后边相邻的select2元素节点。

范例：

|  |
| --- |
| <body > <b>这是一个段落</b> <p id="p1" class="pc1">这是一个段落</p> <b>这是一个段落</b> <b>这是一个段落</b> <b>这是一个段落</b> </body> <script>  $("#p1+b").css("color","red"); </script> |

### 9、兄弟(同辈)元素选择器

语法：$("select1 ~ select2 ")

作用：匹配select1元素节点后面，所有select2的兄弟元素。

范例：

|  |
| --- |
| <body > <b>这是一个段落</b> <p id="p1" class="pc1">这是一个段落</p> <b>这是一个段落</b> <i>这是一个段落</i> <b>这是一个段落</b> </body> <script>  $("#p1~ b").css("color","red"); </script> |

### 10、匹配第一个元素

语法：$("select:first")

作用：匹配第一个select元素节点。

范例：

|  |
| --- |
| <body > <b>这是一个段落</b> <p id="p1" class="pc1">这是一个段落</p> <b>这是一个段落</b> <i>这是一个段落</i> <b>这是一个段落</b> </body> <script>  $("b:first").css("color","red"); </script> |

### 11、匹配最后一个元素

语法：$("select:last")

作用：匹配最后一个select元素节点。

范例：

|  |
| --- |
| <body > <b>这是一个段落</b> <p id="p1" class="pc1">这是一个段落</p> <b>这是一个段落</b> <i>这是一个段落</i> <b>这是一个段落</b> </body> <script>  $("b:last").css("color","red"); </script> |

### 12、去除匹配元素（not）

语法：$("select1:not(select2)")

作用：匹配除了select2 之外的所有select1元素节点。

范例：

|  |
| --- |
| <body > <b>这是1个B</b> <b id="c1">这是2个B</b> <b>这是3个B</b> </body> <script>  $("b:not(#c1)").css("color","red"); </script> |

### 13、匹配索引为偶数的元素（even）

语法：$("select1:even")

作用：匹配索引为偶数的所有select1元素节点。

范例：

|  |
| --- |
| <body > <table>  <tr><td>Header 1</td><td>Header 1</td><td>Header 1</td></tr>  <tr><td>Value 1</td><td>Value 1</td><td>Value 1</td></tr>  <tr><td>Value 2</td><td>Value 2</td><td>Value 2</td></tr>  <tr><td>Value 3</td><td>Value 3</td><td>Value 3</td></tr> </table> </body> <script>  $("tr:even").css("background-color","green"); </script> |

### 14、匹配索引为奇数的元素（odd）

语法：$("select1:odd")

作用：匹配索引为奇数的所有select1元素节点。

范例：

|  |
| --- |
| <body > <table>  <tr><td>Header 1</td><td>Header 1</td><td>Header 1</td></tr>  <tr><td>Value 1</td><td>Value 1</td><td>Value 1</td></tr>  <tr><td>Value 2</td><td>Value 2</td><td>Value 2</td></tr>  <tr><td>Value 3</td><td>Value 3</td><td>Value 3</td></tr> </table> </body> <script>  $("tr:odd").css("background-color","yellow"); </script> |

### 15、匹配指定索引的元素（eq）

语法：$("select1:eq(索引值)")

作用：匹配指定索引的select1元素节点。

范例：

|  |
| --- |
| <body > <table>  <tr><td>Header 1</td><td>Header 1</td><td>Header 1</td></tr>  <tr><td>Value 1</td><td>Value 1</td><td>Value 1</td></tr>  <tr><td>Value 2</td><td>Value 2</td><td>Value 2</td></tr>  <tr><td>Value 3</td><td>Value 3</td><td>Value 3</td></tr> </table> </body> <script>  $("tr:eq(3)").css("background-color","green"); </script> |

### 16、匹配大于指定索引的元素（gt）

语法：$("select1:gt(索引值)")

作用：匹配大于指定索引的select1元素节点。

范例：

|  |
| --- |
| <table>  <tr><td>Header 1</td><td>Header 1</td><td>Header 1</td></tr>  <tr><td>Value 1</td><td>Value 1</td><td>Value 1</td></tr>  <tr><td>Value 2</td><td>Value 2</td><td>Value 2</td></tr>  <tr><td>Value 3</td><td>Value 3</td><td>Value 3</td></tr> </table> </body> <script>  $("tr:gt(3)").css("background-color","green"); </script> |

### 17、匹配小于指定索引的元素（lt）

语法：$("select1:lt(索引值)")

作用：匹配小于指定索引的select1元素节点。

范例：

|  |
| --- |
| <body > <table>  <tr><td>Header 1</td><td>Header 1</td><td>Header 1</td></tr>  <tr><td>Value 1</td><td>Value 1</td><td>Value 1</td></tr>  <tr><td>Value 2</td><td>Value 2</td><td>Value 2</td></tr>  <tr><td>Value 3</td><td>Value 3</td><td>Value 3</td></tr> </table> </body> <script>  $("tr:lt(2)").css("background-color","green"); </script> |

### 18、匹配不可见的元素

语法：$("select1:hidden")

作用：匹配不可见的select1元素节点。

范例：

|  |
| --- |
| <body > <table>  <tr><td>Header 1</td><td>Header 1</td><td>Header 1</td></tr>  <tr style="display: none"><td>Value 1</td><td>Value 1</td><td>Value 1</td></tr>  <tr><td>Value 2</td><td>Value 2</td><td>Value 2</td></tr>  <tr style="display: none"><td>Value 3</td><td>Value 3</td><td>Value 3</td></tr> </table> </body> <script>  $("tr:hidden").css("display","block"); </script> |

### 19、匹配可见的元素

语法：$("select1:visible")

作用：匹配可见的select1元素节点。

范例：

|  |
| --- |
| <body > <table>  <tr>  <td>Header 1</td>  <td>Header 1</td>  <td>Header 1</td></tr>  <tr style="display: none">  <td>Value 1</td>  <td>Value 1</td>  <td>Value 1</td></tr>  <tr>  <td>Value 2</td>  <td>Value 2</td>  <td>Value 2</td></tr>  <tr style="display: none">  <td>Value 3</td>  <td>Value 3</td>  <td>Value 3</td></tr> </table> </body> <script>  $("tr:visible").css("background","red"); </script> |

### 20、匹配带有指定属性的元素

语法：$("select1[属性名]")

作用：匹配带有指定属性的select1元素节点。

范例：

|  |
| --- |
| <body >  账户：<input type="text" name="username" id="username"/><br>  电话：<input type="text" name="phone"/><br>  密码：<input type="password" name="pwd"/><br> </body> <script>  $("input[id]").val("小徐"); </script> |

### 21、匹配指定属性拥有特定值的元素

语法：$("select1[属性名='属性值']")

作用：匹配指定属性拥有特定值的select1元素节点。

范例：

|  |
| --- |
| <body >  账户：<input type="text" name="username"/><br>  电话：<input type="text" name="phone"/><br>  密码：<input type="password" name="pwd"/><br> </body> <script>  $("input[name='username']").val("小徐"); </script> |

### 22、匹配不包含指定属性值的元素

语法：$("select1[属性名!='属性值']")

作用：匹配不包含指定属性值的select1元素节点。

范例：

|  |
| --- |
| <body >  账户：<input type="text" name="username"/><br>  电话：<input type="text" name="phone"/><br>  密码：<input type="password" name="pwd"/><br>  <input type="button"/><br> </body> <script>  $("input[type!='password']").val("小徐"); </script> |

### 23、匹配指定属性以某个值开头的元素

语法：$("select1[属性名^='属性值']")

作用：匹配指定属性以某个值开头的select1元素节点。

范例：

|  |
| --- |
| <body > <p id="p01">段落01</p> <p id="c01">段落01</p> <p id="b01">段落01</p> <p id="p02">段落01</p> </body> <script>  $("p[id^='p']").css("color","red"); </script> |

### 24、匹配指定属性以某个值结尾的元素

语法：$("select1[属性名$='属性值']")

作用：匹配指定属性以某个值结尾的select1元素节点。

范例：

|  |
| --- |
| <body > <p id="p01">段落01</p> <p id="c01">段落01</p> <p id="b01">段落01</p> <p id="p02">段落01</p> </body> <script>  $("p[id$='01']").css("color","red"); </script> |

### 25、匹配指定属性包含某些值的元素

语法：$("select1[属性名\*='属性值']")

作用：匹配指定属性包含某些值的select1元素节点。

范例：

|  |
| --- |
| <body > <p id="and">段落01</p> <p id="cnda">段落01</p> <p id="bab">段落01</p> <p id="pa">段落01</p> </body> <script>  $("p[id\*='a']").css("color","red"); </script> |

### 26、复合属性定位元素

语法：$("select1[属性名\*='属性值'] [属性名^='属性值']...")

作用：根据复合属性及取值来定位指定元素节点。

范例：

|  |
| --- |
| <body > <p id="anda">段落01</p> <p id="cnda">段落01</p> <p id="bab">段落01</p> <p id="pa">段落01</p> </body> <script>  $("p[id^='a'][id$='a']").css("color","red"); </script> |

## 四、HTML内容、文本及值的处理

### 1、操作HTML内容

方法：html([val]);

作用：操作指定元素的HTML内容，如果方法有参数，就代表着更改指定元素的HTML 内容，如果方法中没有参数，就代表着获取指定元素的HTML内容。

范例：

|  |
| --- |
| <body >  <input type="button" value="Input Button"/>  <input type="checkbox" />  <p>这是一段文字  <a href="#" >百度</a>  </p> </body> <script>  $("body").html("<h1>这是一个标题</h1>");  alert($("body").html()); </script> |

### 2、操作HTML元素文本

方法：text([val]);

作用：操作指定元素的文本内容，如果方法有参数，就代表着更改指定元素的文本 内容，如果方法中没有参数，就代表着获取指定元素的文本内容。

范例：

|  |
| --- |
| <body >  <input type="button" value="Input Button"/>  <input type="checkbox" />  <p>这是一段文字  <a href="#" >百度</a>  </p> </body> <script>  $("body").text("<h1>这是一个标题</h1>");  alert($("p").text()); </script> |

注意：html() 方法和text() 方法区别在于，html()方法可以识别html元素。而text()方法它只针对于文本内容。

### 3、操作HTML元素值

方法：val([value]);

作用：操作指定元素的value值，如果方法有参数，就代表着更改指定元素的value值， 如果方法中没有参数，就代表着获取指定元素的value值。

范例：

|  |
| --- |
| <body >  <input type="button" value="Input Button"/>  <input type="text" value="你好"/>  <p>这是一段文字  <a href="#" >百度</a>  </p> </body> <script>  $("input[type='text']").val("<h1>这是一个标题</h1>");  alert($("input[type='text']").val()); </script> |

## 五、HTML属性操作

### 1、获取HTML元素属性值

方法：attr(name);

作用：操作指定元素的name属性，获取指定元素中 name属性的值。

范例：

|  |
| --- |
| <body >  <a href="https://ww.baidu.com" >百度</a> </body> <script>  alert($("a").attr("href")); </script> |

### 2、批量更改HTML元素属性值

方法：attr({属性名:"属性值",属性名:"属性值"...});

作用：操作指定元素的属性值，更改元素中多个属性的值。

范例：

|  |
| --- |
| <body > <a href="https://ww.baidu.com" >百度</a> </body> <script>  alert($("a").attr({href:"https://www.jd.com",target:"\_blank"})); </script> |

### 3、单个更改HTML元素属性值

方法：attr("属性名","属性值");

作用：操作指定元素的属性值，更改指定元素中属性的值。

范例：

|  |
| --- |
| <body > <a href="https://ww.baidu.com" >百度</a> </body> <script>  alert($("a").attr("href","https://www.jd.com")); </script> |

### 4、移除HTML元素属性

方法：removeAttr("属性名");

作用：移除指定元素的属性。

范例：

|  |
| --- |
| <body > <a href="https://ww.baidu.com" >百度</a> </body> <script>  $("a").removeAttr("href"); </script> |

|  |
| --- |
| <body > <a href="https://ww.baidu.com" >百度</a> </body> <script>  //添加class属性  $("a").addClass("abaidu");  $(".abaidu").css("color","red"); </script> |

## 六、class属性值操作

### 1、为元素添加class属性值

方法：addClass("class属性值");

作用：为指定元素动态添加class属性值。

范例：

|  |
| --- |
| <body > <a href="https://ww.baidu.com" >百度</a> </body> <script>  //添加class属性  $("a").addClass("abaidu");  $(".abaidu").css("color","red"); </script> |

### 2、为元素移除class属性值

方法：removeClass("class属性值");

作用：为指定元素移除class属性值。

范例：

|  |
| --- |
| <body > <a href="https://ww.baidu.com" >百度</a> </body> <script>  //添加class属性  $("body").addClass("abaidu");  $("head").addClass("jingdong");  $("html").removeClass("jingdong"); </script> |

### 3、为元素切换class属性值

方法：toggleClass("class属性值");

作用：为指定元素切换class属性值,如果当前元素没有class值就添加，否则就移除。

范例：

|  |
| --- |
| <body > <p class="Hello selected">Hello</p> <p class="again">Hello Again02</p> </body> <script>  //添加class属性  $("p").toggleClass("selected"); </script> |

## 七、HTML样式属性操作

### 1、获取元素指定的样式属性值

方法：css("样式属性名");

作用：通过使用css()方法以及传递的样式属性名，获取对应的样式属性值。

范例：

|  |
| --- |
| <style>  div{  border: 1px solid red;  width:200px;  height: 100px;  }  </style> </head> <body >  <div></div> </body> <script>  **var** borderValue = $("div").css("height");  alert(borderValue); </script> |

### 2、为元素设置单个样式属性

方法：css("样式属性名","样式属性值");

作用：为指定元素设置一个样式属性。

范例：

|  |
| --- |
| <style>  div{  border: 1px solid red;  width:200px;  height: 100px;  }  </style> </head> <body >  <div></div> </body> <script>  $("div").css("border","5px dotted green"); </script> |

### 3、为元素设置多个样式属性

方法：css( { 样式属性名1:"样式属性值1" , 样式属性名2:"样式属性值2"... } );

作用：为指定元素设置多个样式属性。

范例：

|  |
| --- |
| <style>  div{  width:600px;  height:400px;  border:6px solid red;  }  </style> </head> <body >  <div></div> </body> <script>  $("div").css({  width:"200px",  height:"100px",  border:"1px solid green"  }); </script> |

## 八、事件

### 1、页面载入事件

方法：

① $(document).ready(function(){ //js脚本 });

② jQuery(function($){ //脚本 });

③ $(function(){ //脚本 });

作用：在DOM加载完毕之后，再执行js脚本。

范例：

|  |
| --- |
| <script>  //添加页面加载函数，当前文档加载完毕之后，在加载js脚本  $(document).ready(**function** () {  $("div").css({  width:"200px",  height:"100px",  border:"1px solid green"  });  });  </script> </head> <body >  <div></div> </body> |

### 2、失去焦点（光标）事件

方法：blur(function(){ //脚本 })

作用：当前元素失去焦点时，会触发此事件

范例：

|  |
| --- |
| <script>  //添加页面加载函数，当前文档加载完毕之后，在加载js脚本  $(**function**() {  //失去焦点，失去光标  $("input").blur(**function**(){  //this 表示的是当前对象， $(this)当前元素节点  $(**this**).val("猪");  });  });  </script> </head> <body >  你是<input type="text"> </body> |

### 3、改变值事件

方法：change(function(){ //脚本 })

作用：当前元素值发生了改变并失去焦点时，会触发此事件

范例：

|  |
| --- |
| <script>  //添加页面加载函数，当前文档加载完毕之后，在加载js脚本  $(**function**() {  //改变值后失去焦点，触发此事件  $("input").change(**function**(){  //this 表示的是当前对象， $(this)当前元素节点  $(**this**).val("猪");  });  });  </script> </head> <body >  你是<input type="text"> </body> |

### 4、鼠标双击事件

方法：dblclick(function(){ //脚本 })

作用：鼠标双击当前元素时，会触发此事件

范例：

|  |
| --- |
| <script>  //添加页面加载函数，当前文档加载完毕之后，在加载js脚本  $(**function**() {  $("div").dblclick(**function** () {  alert("鼠标双击事件");  });  });  </script> </head> <body >  <div>双击点我啊！变变变</div> </body> |

### 5、获取焦点事件

方法：foucs(function(){ //脚本 })

作用：当元素获取到焦点时，会触发此事件

范例：

|  |
| --- |
| <script>  //添加页面加载函数，当前文档加载完毕之后，在加载js脚本  $(**function**() {  $("input[type='text']").focus(**function** () {  alert("获取焦点事件");  **this**.blur();  });  });  </script> </head> <body >  账户：<input type="text"> </body> |

### 6、鼠标移入事件

方法：mouseover(function(){ //脚本 })

作用：当鼠标移入到元素时，会触发此事件

范例：

|  |
| --- |
| <script>  //自动加载函数，页面载入，此函数会在当前文档加载完毕之后，在加载执行  $(**function**(){  //鼠标移入事件  $("input[type='button']").mouseover(**function**(){  $("span").text("鼠标移入，文本再现。");  });   });  </script> </head> <body >  <input type="button" value="提示信息">  <span ></span> </body> |

### 7、鼠标移出事件

方法：mouseout(function(){ //脚本 })

作用：当鼠标移出当前元素时，会触发此事件

范例：

|  |
| --- |
| <script>  //自动加载函数，页面载入，此函数会在当前文档加载完毕之后，在加载执行  $(**function**(){  //鼠标移出事件  $("input[type='button']").mouseout(**function**(){  $("span").text("鼠标移出，二娇以离开。");  });  });  </script> </head> <body >  <input type="button" value="提示信息">  <span ></span> </body> |

### 8、鼠标移动事件

方法：mousemove(function(){ //脚本 })

作用：当鼠标在当前元素上移动时，会触发此事件

范例：

|  |
| --- |
| <script>  //自动加载函数，页面载入，此函数会在当前文档加载完毕之后，在加载执行  $(**function**(){  //鼠标移入事件  $("input[type='button']").mouseover(**function**(){  alert("移入");  /\*$("span").text("鼠标移入，文本再现。");\*/  });  //鼠标移动事件  $("input").mousemove(**function**(){  /\* $("span").text("鼠标移动，亮晶晶！");\*/  alert("移动");  });  });  </script> </head> <body >  <input type="button" value="提示信息">  <span ></span> </body> |

### 9、鼠标点击时事件

方法：mousedown(function(){ //脚本 })

作用：当鼠标点击时，会触发此事件，即使鼠标没有松开。

注意：与click()的区别在于，click事件需要鼠标点击后松开鼠标，才会触发。

而mousedown会在鼠标点击时就触发，即使鼠标没松开。

范例：

|  |
| --- |
| <script>  //自动加载函数，页面载入，此函数会在当前文档加载完毕之后，在加载执行  $(**function**(){  //鼠标点击时事件（鼠标点击，没有松开时就会触发此事件）  $("input").mousedown(**function**(){  $("span").text("鼠标点击时，文本再现。");  });  });  </script> </head> <body >  <input type="button" value="提示信息">  <span ></span> </body> |

### 10、鼠标点击后松开时事件

方法：mouseup(function(){ //脚本 })

作用：当鼠标点击元素松开鼠标时，会触发此事件。

范例：

|  |  |
| --- | --- |
| |  | | --- | | <script>  //自动加载函数，页面载入，此函数会在当前文档加载完毕之后，在加载执行  $(**function**(){  //鼠标点击时事件（鼠标点击，没有松开时就会触发此事件）  $("input").mousedown(**function**(){  $("span").text("鼠标点击，文本再现。");  });  $("input").mouseup(**function**(){  $("span").text("鼠标松开，文本不在。");  });  });  </script> </head> <body >  <input type="button" value="提示信息">  <span ></span> </body> | |

## 九、效果

### 1、show() 显示元素

方法：show("slow","normal","fast"|毫秒数值)

作用：显示元素，如果没有参数，没有效果直接显示。如果有参数，根据参数来确 定缓冲的速度。

范例：

|  |
| --- |
| <script>  //自动加载函数 $(**function**(){  //鼠标点击 $(".one li div").click(**function** () {  **var** display = $(**this**).next().css("display");  **if**(display=="none"){  $(**this**).next().show(3000);  }**else**{  $(**this**).next().css("display","none");  } });  });  </script> </head> <body >  <!--设置一个最外层div容器-->  <div class="wrap">  <!--添加一个无序列表-->  <ul class="one">  <li>  <div>一级菜单03</div>  <ul class="two">  <li>一级菜单01</li>  <li>一级菜单02</li>  <li>一级菜单03</li>  <li>一级菜单04</li>  </ul>  </li>  </ul>  </div> </body> |

### 2、hide() 隐藏元素

方法：hide("slow","normal","fast"|毫秒数值)

作用：隐藏元素，如果没有参数，没有效果直接隐藏。如果有参数，根据参数来确 定缓冲的速度。

范例：

|  |
| --- |
| <script>  //自动加载函数  $(**function**(){ //鼠标点击  $(".one li div").click(**function** () {  **var** display = $(**this**).next().css("display");  **if**(display=="none"){  $(**this**).next().show(600);  }**else**{  $(**this**).next().hide(600);  }  });  }); </script> </head> <body > <!--设置一个最外层div容器--> <div class="wrap">  <!--添加一个无序列表-->  <ul class="one">  <li>  <div>一级菜单03</div>  <ul class="two">  <li>一级菜单01</li>  <li>一级菜单02</li>  <li>一级菜单03</li>  <li>一级菜单04</li>  </ul>  </li>  </ul> </div> </body> |

### 3、toggle() 显示与隐藏元素

方法：toggle("slow","normal","fast"|毫秒数值)

作用：如果元素的状态是隐藏，那么就显示元素；如果元素的状态是显示，那么就隐藏 元素。有参数就有缓冲的效果。没有参数，就没有缓冲效果。

范例：

|  |
| --- |
| <script>  //自动加载函数  $(**function**(){ //鼠标点击  //鼠标点击  $(".one li div").click(**function** () {  $(**this**).next().toggle(600);  //让当前一级菜单元素改变样式  $(**this**).css({  backgroundColor:"#fedf7c",  borderColor:"#C3262D",  color:"#7f031b"  });  });  }); </script> </head> <body > <!--设置一个最外层div容器--> <div class="wrap">  <!--添加一个无序列表-->  <ul class="one">  <li>  <div>一级菜单03</div>  <ul class="two">  <li>一级菜单01</li>  <li>一级菜单02</li>  <li>一级菜单03</li>  <li>一级菜单04</li>  </ul>  </li>  </ul> </div> </body> |

### 4、slideDown() 元素滑下

方法：slideDown("slow","normal","fast"|毫秒数值)

作用：缓慢的将元素滑下。

范例：

|  |
| --- |
| <script>  //自动加载函数  $(**function**(){ //鼠标点击  //鼠标点击  $(".one li div").click(**function** () {  $(**this**).next().slideDown(800);  //让当前一级菜单元素改变样式  $(**this**).css({  backgroundColor:"#fedf7c",  borderColor:"#C3262D",  color:"#7f031b"  });  });  }); </script> </head> <body > <!--设置一个最外层div容器--> <div class="wrap">  <!--添加一个无序列表-->  <ul class="one">  <li>  <div>一级菜单03</div>  <ul class="two">  <li>一级菜单01</li>  <li>一级菜单02</li>  <li>一级菜单03</li>  <li>一级菜单04</li>  </ul>  </li>  </ul> </div> </body> |

### 5、slideUp() 元素滑上

方法：slideUp("slow","normal","fast"|毫秒数值)

作用：缓慢的将元素滑上。

范例：

|  |
| --- |
| <script>  //自动加载函数  $(**function**(){ //鼠标点击  //鼠标点击  $(".one li div").click(**function** () {  **var** display = $(**this**).next().css("display");  **if**(display=="none"){  $(**this**).next().slideDown(600);  }**else**{  $(**this**).next().slideUp(600);  }  //让当前一级菜单元素改变样式  $(**this**).css({  backgroundColor:"#fedf7c",  borderColor:"#C3262D",  color:"#7f031b"  });  });  }); </script> </head> <body > <!--设置一个最外层div容器--> <div class="wrap">  <!--添加一个无序列表-->  <ul class="one">  <li>  <div>一级菜单03</div>  <ul class="two">  <li>一级菜单01</li>  <li>一级菜单02</li>  <li>一级菜单03</li>  <li>一级菜单04</li>  </ul>  </li>  </ul> </div> </body> |

### 6、slideToggle() 元素滑上或滑下

方法：slideToggle("slow","normal","fast"|毫秒数值)

作用：缓慢的将元素滑上或滑下。如果元素的状态是显示，就将元素缓慢的滑上；如果 元素的状态是隐藏，就将元素缓慢的滑下。

范例：

|  |
| --- |
| <script>  //自动加载函数  $(**function**(){ //鼠标点击  //鼠标点击  $(".one li div").click(**function** () {  $(**this**).next().slideToggle(600);  //让当前一级菜单元素改变样式  $(**this**).css({  backgroundColor:"#fedf7c",  borderColor:"#C3262D",  color:"#7f031b"  });  });  }); </script> </head> <body > <!--设置一个最外层div容器--> <div class="wrap">  <!--添加一个无序列表-->  <ul class="one">  <li>  <div>一级菜单03</div>  <ul class="two">  <li>一级菜单01</li>  <li>一级菜单02</li>  <li>一级菜单03</li>  <li>一级菜单04</li>  </ul>  </li>  </ul> </div> </body> |

### 7、fadeIn() 元素淡入

方法：fadeIn("slow","normal","fast"|毫秒数值)

作用：元素淡入。

范例：

|  |
| --- |
| <script>  //自动加载函数  $(**function**(){  //鼠标点击  $(".one li div").click(**function** () {  **var** display = $(**this**).next().css("display");  **if**(display=="none"){  $(**this**).next().fadeIn(3000);  }**else**{  $(**this**).next().css("display","none");  }  //让当前一级菜单元素改变样式  $(**this**).css({  backgroundColor:"#fedf7c",  borderColor:"#C3262D",  color:"#7f031b"  });  });  }); </script> </head> <body > <!--设置一个最外层div容器--> <div class="wrap">  <!--添加一个无序列表-->  <ul class="one">  <li>  <div>一级菜单01</div>  <ul class="two">  <li>一级菜单01</li>  <li>一级菜单02</li>  <li>一级菜单03</li>  <li>一级菜单04</li>  </ul>  </li>  </ul> </div> </body> |

### 8、fadeOut() 元素淡出

方法：fadeOut("slow","normal","fast"|毫秒数值)

作用：元素淡出。

范例：

|  |
| --- |
| <script>  //自动加载函数  $(**function**(){  //鼠标点击  $(".one li div").click(**function** () {  **var** display = $(**this**).next().css("display");  **if**(display=="none"){  $(**this**).next().fadeIn(3000);  }**else**{  $(**this**).next().fadeOut(3000);  }  //让当前一级菜单元素改变样式  $(**this**).css({  backgroundColor:"#fedf7c",  borderColor:"#C3262D",  color:"#7f031b"  });  });  }); </script> </head> <body > <!--设置一个最外层div容器--> <div class="wrap">  <!--添加一个无序列表-->  <ul class="one">  <li>  <div>一级菜单01</div>  <ul class="two">  <li>一级菜单01</li>  <li>一级菜单02</li>  <li>一级菜单03</li>  <li>一级菜单04</li>  </ul>  </li>  </ul> </div> </body> |

### 9、fadeTo() 元素淡入(指定不透明度)

方法：fadeOut("slow","normal","fast"|毫秒数值,不透明度[0~1])

作用：元素淡出,可以指定元素的不透明度。

范例：

|  |
| --- |
| <script>  //自动加载函数  $(**function**(){  //鼠标点击  $(".one li div").click(**function** () {  **var** display = $(**this**).next().css("display");  **if**(display=="none"){  $(**this**).next().fadeTo(1000,1);  }**else**{  $(**this**).next().fadeOut(3000);  }  //让当前一级菜单元素改变样式  $(**this**).css({  backgroundColor:"#fedf7c",  borderColor:"#C3262D",  color:"#7f031b"  });  });  }); </script> </head> <body > <!--设置一个最外层div容器--> <div class="wrap">  <!--添加一个无序列表-->  <ul class="one">  <li>  <div>一级菜单01</div>  <ul class="two">  <li>一级菜单01</li>  <li>一级菜单02</li>  <li>一级菜单03</li>  <li>一级菜单04</li>  </ul>  </li>  </ul> </div> </body> |

### 10、fadeToggle() 元素淡入或淡出

方法：fadeToggle("slow","normal","fast"|毫秒数值)

作用：元素淡入或淡出,如果元素的状态是显示，就淡出；如果元素的状态是隐藏，就 淡入。

范例：

|  |
| --- |
| <script>  //自动加载函数  $(**function**(){  //鼠标点击  $(".one li div").click(**function** () {  $(**this**).next().fadeToggle(600);  //让当前一级菜单元素改变样式  $(**this**).css({  backgroundColor:"#fedf7c",  borderColor:"#C3262D",  color:"#7f031b"  });  });  }); </script> </head> <body > <!--设置一个最外层div容器--> <div class="wrap">  <!--添加一个无序列表-->  <ul class="one">  <li>  <div>一级菜单01</div>  <ul class="two">  <li>一级菜单01</li>  <li>一级菜单02</li>  <li>一级菜单03</li>  <li>一级菜单04</li>  </ul>  </li>  </ul> </div> </body> |

## 十、迭代方法

在使用jquery 选择元素时，可以选择多个元素节点，

|  |
| --- |
| */\*迭代元素的方法 ： each\*/ $("p").each(function () {  var text = $(this).text();  alert(text); })*  <**p**>第一个段落</**p**> <**p**>第二个段落</**p**> <**p**>第三个段落</**p**> |