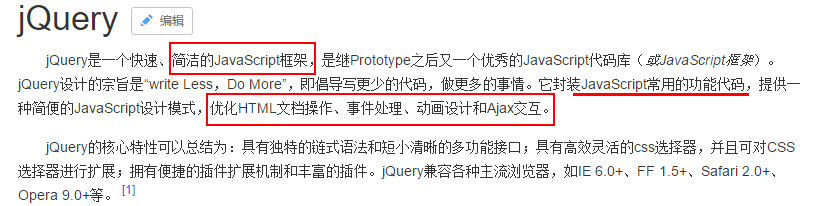
|  |
| --- |
|  |

JQuery

# 什么是JQuery

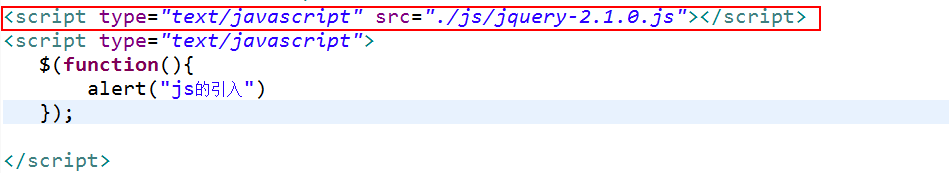


# Jquery的使用方法

Jquery它是一个库(框架)，相当于java中能够的jar包，要想使用它，必须先引入！

jquery-1.8.3.js：一般用于学习，可读性强

jquery-1.8.3.min.js：用于项目使用阶段，可读性极差



注意：引入js库的<script></script>标签内部不允许写js代码，即使写了也不执行 。

# jQuery的快速入门

## jQuery快速入门

### 页面加载函数

所有的jquery代码写在页面加载函数（页面加载完成执行这里面的代码，避免操作元素时未加载的问题）

$(function(){

Jquery代码

});

### Jquery获取元素与原始js获取元素对比

|  |
| --- |
| //原始js获取审判元素  document.getElementById("span").innerHTML="原始的js操作" ;  //jQuery操作  $("#span").html("jQuery操作"); |

**优点：简化操作**

### 原始的dom对象和jQuery对象之间的转换

#### 获取原始的dom对象

|  |
| --- |
| //获取原始的dom对象 spa  **var** spa = document.getElementById("span");  spa.innerHTML="spa是一个原始的dom对象" ; |

#### 获取jQuery对象

|  |
| --- |
| **var** $spa = $("#span") ;  //jQuery操作  $spa .html("$spa是一个jQuery对象"); |

#### 原始的dom对象转换成jQuery对象

原始dom对象转换成jQuery对象以后可以操作jQuery定义的方法

|  |
| --- |
| //获取原始的dom对象 spa  **var** spa = document.getElementById("span");  spa.innerHTML="spa是一个原始的dom对象" ;  //将原始的dom对象转换成jQuery对象  $(spa).html("spa就变成了一个JQuery对象") ; |

#### jQuery对象转换成原始的dom对象（没卵用）

|  |
| --- |
| **var** $spa = $("#span") ;  //jQuery操作  $spa.get(0).innerHTML="转换成bom对象" ; |

**注意：JQ对象只能操作JQ里面的属性和方法**

**JS对象只能操作JS里面的属性和方法。**

# jQuery的选择器

## 什么是选择器，

可认为是之前学的CSS的选择器。准确定位，选择具体某个、某些元素

## 基本选择器

* **id选择器：$(“#id名称”);**
* **元素选择器：$(“元素名称”);**
* **类选择器：$(“.类名”);**
* **通配符：\***
* **多个选择器共用(并集)**

|  |
| --- |
| //选择id为one的元素，将其样式背景改为粉色CSS();是一个方法  $("#one").css("background-color","pink");  //选择class类为mini的元素，  $(".mini").css("background-color","pink");  //选择元素div  $("div").css("background-color","pink");  //选择所有的元素  $("\*").css("background-color","pink");  //多选择器 选择id为two的元素和class类为mini的元素  $("#two .mini").css("background-color","pink"); |

## 层级选择器



**ancestor descendant: 在给定的祖先元素下匹配所有的后代元素**

**parent > child : 在给定的父元素下匹配所有的子元素**

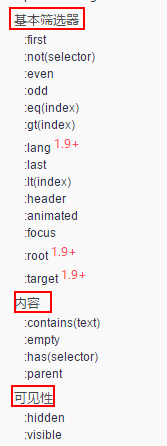
**prev + next: 匹配所有紧接在 prev 元素后的 next 元素(紧挨着的一个)**

**prev ~ siblings: 匹配 prev 元素之后的所有 siblings 元素(兄弟)**

|  |
| --- |
| <script>  $(**function**(){  //给ID为btn1 的按钮绑定一个点击事件  $("#btn1").click(**function**(){  //元素body下所有的div 只要是div就会选中  $("body div").css("background-color","pink");  });  $("#btn2").click(**function**(){  //自会选中body的左右的子div id为 one two tree four 的div  $("body>div").css("background-color","pink");  });  $("#btn3").click(**function**(){  //会选中id为one的元素 和紧挨着它的兄弟元素 也就是id为two的div  $("#one + div").css("background-color","pink");  });  $("#btn4").click(**function**(){  //后面所有的兄弟元素 不包括自己  $("#one ~ div").css("background-color","pink");  });    });  </script>  </head>  <body>  <input type=*"button"* id=*"btn1"* value=*"选择body中的所有的div元素"*/>  <input type=*"button"* id=*"btn2"* value=*"选择body中的第一级的孩子"*/>  <input type=*"button"* id=*"btn3"* value=*"选择id为two的元素的下一个元素"*/>  <input type=*"button"* id=*"btn4"* value=*"选择id为one的所有的兄弟元素"*/>    <hr/>  <div id=*"one"*>  <div class=*"mini"*>  第一块  </div>  </div>    <div id=*"two"*>  <div class=*"mini"*>  第二块  </div>  <div class=*"mini"*>  第三块  </div>  </div>    <div id=*"three"*>  <div class=*"mini"*>  第四块  </div>  <div class=*"mini"*>  第五块  </div>  <div class=*"mini"*>  第六块  </div>  </div>    <span id=*"four"*>    </span>  </body> |

## 基本筛选器

都是以 **:** 开头 筛选出



|  |
| --- |
| $("div:last").css("background-color","red");  $("div:first").css("background-color","red");    $("div:odd").css("background-color","red");    $("div:even").css("background-color","red"); |

## 表单选择器



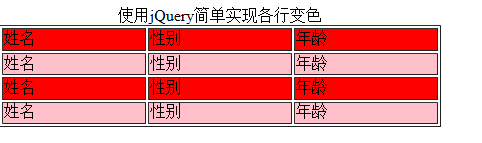
|  |
| --- |
| $(":input")  $(":password") |

很简单不做过多演示。

## 案例

### 案例一 简单实现各行变色

#### 实现结果



#### 实现代码

|  |
| --- |
| <script>  $(**function**(){  $("table tr:even").css("background-color","red");  $("table tr:odd").css("background-color","pink");    });  </script>  </head>  <body>  <table border=*"1"*>  <caption>使用jQuery简单实现各行变色</caption>  <tr>  <td width=*"140"*>姓名</td>  <td width=*"140"*>性别</td>  <td width=*"140"*>年龄</td>  </tr>  <tr>  <td width=*"140"*>姓名</td>  <td width=*"140"*>性别</td>  <td width=*"140"*>年龄</td>  </tr>  <tr>  <td width=*"140"*>姓名</td>  <td width=*"140"*>性别</td>  <td width=*"140"*>年龄</td>  </tr>  <tr>  <td width=*"140"*>姓名</td>  <td width=*"140"*>性别</td>  <td width=*"140"*>年龄</td>  </tr>  </table>  </body> |

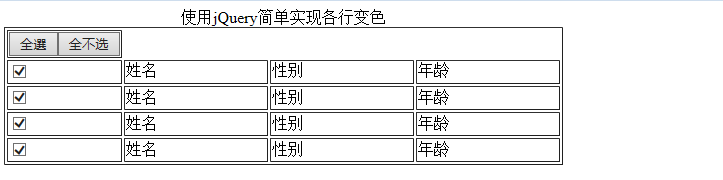
#### 总结

**主要使用到元素选择器，层级选择器 ，筛选器**

$("table tr:even") 元素table下的所有的tr元素 、 帅选出奇、偶元素

### 案例二 实现全选不选

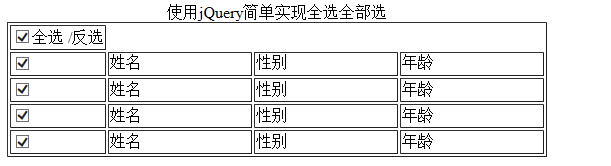
#### 效果一



#### 实现代码

|  |
| --- |
| <script>  $(**function**(){  $("#btn1").click(**function**(){  //表单选择器  $(":checkbox").prop("checked",**true**);  //只起作用一次  //$(":checkbox").attr("checked",true);  });  $("#btn2").click(**function**(){  $(":checkbox").prop("checked",**false**);  });  });  </script> |

#### 效果二



|  |
| --- |
| <script>  $(**function**(){  $("#all").click(**function**(){  $(":checkbox").prop("checked",**this**.checked);  });    });  </script>  </head>  <body>  <table border=*"1"*>  <caption>使用jQuery简单实现全选全部选</caption>  <tr>  <td><input id=*"all"* type=*"checkbox"* />全选 /反选 </td>  </tr>  <tr>  <td><input type=*"checkbox"*/></td>  <td width=*"140"*>姓名</td>  <td width=*"140"*>性别</td>  <td width=*"140"*>年龄</td>  </tr>  <tr>  <td><input type=*"checkbox"*/></td>  <td width=*"140"*>姓名</td>  <td width=*"140"*>性别</td>  <td width=*"140"*>年龄</td>  </tr>  <tr>  <td><input type=*"checkbox"*/></td>  <td width=*"140"*>姓名</td>  <td width=*"140"*>性别</td>  <td width=*"140"*>年龄</td>  </tr>  <tr>  <td><input type=*"checkbox"*/></td>  <td width=*"140"*>姓名</td>  <td width=*"140"*>性别</td>  <td width=*"140"*>年龄</td>  </tr>  </table>  </body> |

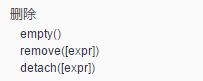
#### 总结

还是使用基本选择器，明确要操作的元素及具体属性

# 文档处理



|  |
| --- |
| <script type=*"text/javascript"*>  $(**function**(){  //添加到后面  $("#btn1").click(**function**(){  $("#bj").appendTo("#love");  });  $("#btn2").click(**function**(){  $("#love").append($("#bj"));  });  //追加到前面  $("#btn3").click(**function**(){  $("#love").prepend($("#cs"));  });  $("#btn4").click(**function**(){  $("#cs").prependTo("#love");  });  });  </script>  </head>  <body>  <ul id=*"city"*>  <li id=*"bj"* name=*"bj"*>北京</li>  <li id=*"tj"* name=*"tj"*>天津</li>  <li id=*"cs"* name=*"cs"*>长沙</li>  </ul>  <hr border=*"10"*/>  <ul id=*"love"*>  <li id=*"lol"* name=*"lol"*>撸啊撸</li>  <li id=*"cf"* name=*"cf"*>CROSSFIRE</li>  </ul>    <button id=*"btn1"*>appendTo</button>  <button id=*"btn2"*>appen</button>  <button id=*"btn3"*>prepend</button>  <button id=*"btn4"*>prependTo</button>  </body> |



empty() 清空标签体

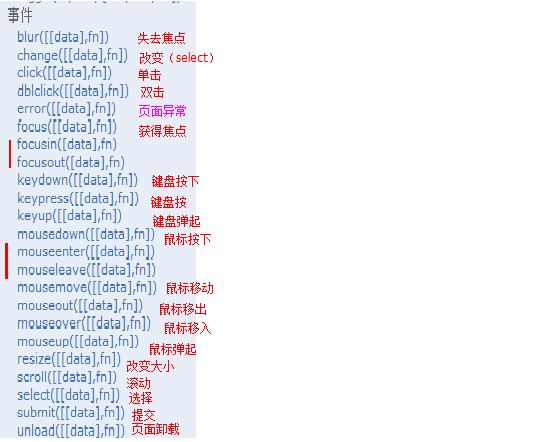
remove() 删除当前对象。如果之后再使用，元素本身保留，绑定事件 或 绑定数据 都会被移除

detach() 删除当前对象。如果之后再使用，元素本身保留，绑定事件 或 绑定数据 都保留

# 事件

## 原生js事件和jQuery事件比较

jQuery中的事件，是对原生的js事件的补充和完善 。



Js中的事件



比较发现规律，原生的js事件前面所有on 。

## 常用事件

focusin 和 focusout

focusin 获得焦点。js focus。

focusin事件跟focus事件区别在于，他可以在父元素上检测子元素获取焦点的情况。

focusout 失去焦点。js blur。

focusout事件跟blur事件区别在于，他可以在父元素上检测子元素失去焦点的情况。

mouseenter 和 mouseleave

mouseenter 鼠标移入。js mouseover

与 mouseover 事件不同，只有在鼠标指针穿过被选元素时，才会触发 mouseenter

事件。如果鼠标指针穿过任何子元素，同样会触发 mouseover 事件。

mouseleave 鼠标移出。js mouseout

与 mouseout 事件不同，只有在鼠标指针离开被选元素时，才会触发 mouseleave

事件。如果鼠标指针离开任何子元素，同样会触发 mouseout 事件。

|  |
| --- |
| <script type=*"text/javascript"*>  $(**function**(){  $("#name").blur(**function**(){  $("#textMsg").html("输入文本框失去焦点事件----blur(function(){})");  });  $("#name").focus(**function**(){  $("#textMsg").html("输入文本框获得焦点事件----focus(function(){})");  });  </script>  </head>  <body>  姓名:<input id=*"name"* type=*"text"* /><span id=*"textMsg"*></span> <br/>  </body> |

不过多演示 。

# 绑定事件

**给某个元素绑定一个事件，使其具有特定的功能（触发方法）。**



## on**在选择元素上绑定一个或多个事件的事件处理函数。**



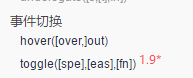
## one**为每一个匹配元素的特定事件（像click）绑定一个一次性的事件**

|  |
| --- |
| $(**function**(){  $("#bnt").one("click", **function**(){  alert("one 只能执行一次这个方法");  });  }) |

## off **在选择元素上移除一个或多个事件的事件处理函数**

|  |
| --- |
| $("#bnt").off("click") ;//解绑bnt的点击事件 |

# 事件切换



## **Hover**

**hover(over , out)**

**简化版，鼠标移入和移出 ，A.mouseover( fn ).mouseout( fn) 简化 A.hover( fn , fn )**

**鼠标移进执行一个方法，移出执行一个方法 。**

## Toggle

toggle( fn , fn , fn ,...) 执行click事件，每点击一次执行一个fn 事件 ，循环执行