## [jQuery设置disabled属性与移除disabled属性](https://www.cnblogs.com/73tong/p/7922099.html)

**表单中readOnly和disabled的区别：**

1、Readonly只针对input(text/ password)和textarea有效，而disabled对于所有的表单元素都有效，包括select,radio, checkbox, button等。

2、但是表单元素在使用了disabled后，当我们将表单以POST或GET的方式提交的话，这个元素的值不会被传递出去，而readonly会将该值传递出去（这种情况出现在我们将某个表单中的textarea元素设置为disabled或readonly，但是submitbutton却是可以使用的）。

**jQuery进行操作：**

//两种方法设置disabled属性

$('#areaSelect').attr("disabled",true);

$('#areaSelect').attr("disabled","disabled");

//三种方法移除disabled属性

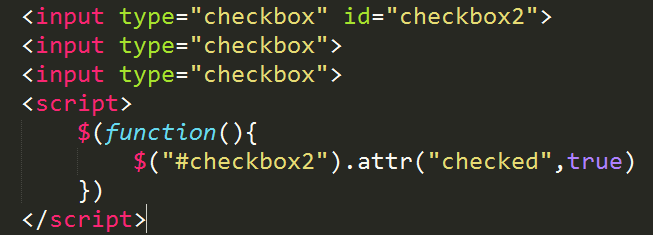
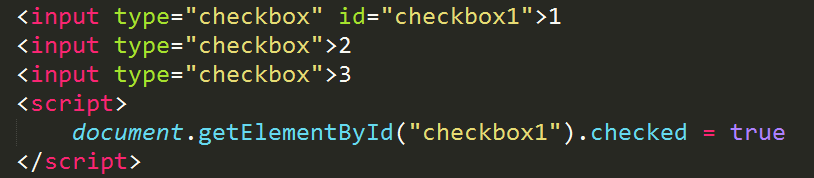
$('#areaSelect').attr("disabled",false);

$('#areaSelect').removeAttr("disabled");

$('#areaSelect').attr("disabled","");

## **[复选框之checked属性](https://www.cnblogs.com/jf-67/p/6613898.html)**

复选框里只要有checked属性，不管是否为其赋值，结果为空或true或false或任意值，均为选中状态。



**无论是用javascript还是jQuery来设置checked属性，其值均为true或false，切忌带引号，否则会出错。**

**在js中，我们可以这样来写：**

var checkbox1 = document.getElementById("checkbox1");

var checkbox2 = document.getElementById("checkbox2");

var checkbox3 = document.getElementById("checkbox3");

console.log(checkbox1.checked); // true

console.log(checkbox2.checked)  // false

console.log(checkbox3.checked)  // true

**在jQuery中，可以这样获取：**

$(function(){

      console.log($("#checkbox1").attr('checked'))  // checked

      console.log($("#checkbox2").attr('checked'))  // undefined

      console.log($("#checkbox3").attr('checked'))  // checked

})

　　从上面例子可以看出同样是获取复选框是否选中，js的获取值是布尔值，即true 或者 false，而jQuery的获取值则是checked或者undefined。

阻止a链接跳转方法

**方法一：**<a href="javascript:void(0)">百度</a>

**方法二：**用return false

<a href="http://www.baidu.com" onclick=" myjs(); return false;">百度 </a>

<a href="http://www.baidu.com" onclick=" return false; ">百度</a>

js中数据类型判断

1.typeof

用来判断各种数据类型， typeof()

2. instanceof

判断已知对象类型的方法。变量 instancof （Array/Date/Function），instanceof 后面一定要是对象类型，并且大小写不能错。

3.constructor

根据对象的constructor===（Array/Date/Function）判断，返回对创建此对象的数组函数的引用。

4.prototype

所有数据类型均可判断：Object.prototype.toString.call(变量)，输出数据类型。

**js中判断变量是否为空：**

**if**(value == **null** || value == "" || value == "undefined" || value == undefined || value == "null"){

**return** **true**;

    }

[confirm对话框取消后阻止ajax操作、ajax做批量删除](https://www.cnblogs.com/xiaofox0018/p/6244727.html)

在做批量删除的时候，需要用confirm弹出一个提示框让用户确认是否删除，点击确定，执行操作，点击取消，取消操作。但是如果使用ajax把选中项的主键值传到处理页面处理时，如果使用下面的方法将confirm直接写在按钮里面，在点击取消后，ajax会继续执行操作，删除选中项。

**1.confirm错误写法：**

<button type="button" onclick="return confirm('确定删除选中项吗！')" id="piliang" class='btn btn-primary btn-lg' data-target='#myModal'>批量删除</button>

**2.可以用下面的方法将confirm写在js里面，可以避免这个问题。**

**（1）按钮代码**

<button type="button" id="piliang" class='btn btn-primary btn-lg' data-target='#myModal'>批量删除</button>

**（2）js代码**

$("#piliang").click(function(){

var statu = confirm("确认删除选中项吗！");//在js里面写confirm，在页面中弹出提示信息。

if(!statu)//如果点击的是取消

{

return false;//返回页面

}

else{//如果点击确定，就继续执行下面的操作

var attr = new Array();//建立一个数组

$(".qx").each(function() {//each循环遍历出所有的复选框

if($(this).prop("checked")) //判断被循环的.ck是否被选中

{

attr.push($(this).attr("value")); //去value的属性取到该项数据的主键值,扔到数组attr里面

}

});

var stra = attr.join("','");//把attr数组里的lie[0]用 ',' 拆分

$.ajax({

url:"../chuli/deletepiliang.php",

data:{stra:stra},//把拆分后的数组传到处理页面，数据的格式为 主键1','主键2','主键3 这里看似少了最外面的2个''，带着这个疑问去看处理页面

type:"POST",

dataType:"TEXT",

success: function(data){

window.location.href="guanliry.php";

}

});

}

})

**（3）处理页面deletepiliang.php**

<?php

session\_start();

include("../libs/DBDA.php");

$db = new DBDA();

$attr = ($\_POST["stra"]);//取传过来的选定的主键值

$sql = "delete from jx\_user where uid in('{$attr}')";//把上面的缺少最外面的''的字符串放到这里，('{$attr}')中的{$attr}就是拿过来的字符串，外面的''正好弥补了缺少的''。

$db->query($sql,0);

JavaScript：

var arr=[值1,值2,值3];                     //隐式创建

var arr=new Array(值1,值2,值3);      //直接实例化

var array=new Array(size);           //创建数组并指定长度

array['key'] = value                      //常用

**[jquery将table的td转化成可编辑的文本](https://www.cnblogs.com/hwaggLee/p/5192886.html)**

**代码一：**

function editTable\_t(){

$("table").on("click", "td:not(.active)", function () {

var $this = $(this);

var $textbox = $("<input>", { type: "text", size: 5, value: $this.addClass("active").text() });

$this.html($textbox);

$textbox.focus();

});

$("table").on("blur", "input:text", function () {

var $this = $(this);

$this.parent().removeClass("active").text($this.val());

});

}

**代码二：**

function editTable\_t(){

$("table tr td").attr("contenteditable", true);

$("table tr td").each(function() {

var contents = $(this).html();

$(this).html($("<div>").append(contents));

});

$("table tr td div").attr("contenteditable", true);

}

**代码三：**

function editTable\_t(){

$("table td").click( function( e ){

if ( $(this).find('input').length ) {

return ;

}

var input = $("<input type='text' size='5' />").val( $(this).text() );

$(this).empty().append( input );

$(this).find('input') .focus().blur( function( e ){

$(this).parent('td').text(

$(this).val()

);

});

});

}

**代码四：**

$("table tr td").on('blur',"input[type='text']", function( e ){

$(this).closest('td').text(

$(this).val()

);

});

$("table").on('click','td', function( e ){

if ( $(this).find('input').length ) {

return ;

}

var input = $("<input type='text' size='5' />").val( $(this).text() );

$(this).empty().append( input );

});

**代码五：**

var z={};

function tdClicks(){

var x="",y="";

$("table tr td").click(function(){

z=$(this);

x = $(this).text() || $(this).find("input[type='text']").val();

if(!x){

x="";

}

$(this).html("<input type='text' size='5' value='"+ x+"' />");

$(this).unbind("click");

$(this).find("input[type='text']").bind("blur", function(){

catchme($(this).val());

tdClicks();

});

});

}

function catchme(wht){

$(z).text(wht);

}

tdClicks();