



黑马程序员
www.itheima.com

传智播客旗下
高端IT教育品牌

jQuery

讲师：小杨



课程介绍:红色标记为重点内容



课前说明

内容：掌握jQuery编程思想，使用jQuery进行常见网页效果开发。

目标：能够使用jQuery开发常见网页效果。

参考书：《锋利的jQuery》

jQuery官网：<http://jquery.com>

jQuery在线API：<http://api.jquery.com>
<http://api.jquery.com/api/> (xml文件。)

jQuery UI：<http://jqueryui.com/>

常见的JavaScript框架库

什么是JavaScript框架库？

普通JavaScript的缺点：每种控件的操作方式不统一，不同浏览器下有区别封装库，要编写跨浏览器的程序非常麻烦。因此出现了很多对JavaScript的。

常见的JavaScript框架库

Prototype、YUI(网络反响一般)、Dojo、ExtJS、jQuery等，这些库对JavaScript进行了封装，简化了开发。这些库是对JavaScript的封装，内部都是用JavaScript实现的。

jQuery:

jQuery就是JavaScript语法写的一些函数类，内部仍然是调用JavaScript实现的，所以**并不是代替JavaScript的**。使用jQuery的代码、编写jQuery的扩展插件等仍然需要JavaScript的技术，jQuery本身就是一堆JavaScript函数。

jQuery简介

什么是jQuery?

jQuery就是一个JavaScript函数库，没什么特别的。

jQuery能做什么？jQuery是做什么的？

jQuery本身就是一堆JavaScript函数，JavaScript是做什么的，jQuery也一样。毕竟jQuery只是用JavaScript编写的函数库而已，有些功能jQuery没有封装，则还需要通过自己写JavaScript来实现。

为什么要学习jQuery

Js中一个简单的功能需要大量的代码

Js中兼容的问题很多

JS中代码的容错性很差

Js中window.onload也只能有一个

。
。 。 。 。 。 。 。 。 。

问题比较多，总之学习JQ之后你会感到自己
从地狱升到了天堂,而且企业里都用这玩意

好处

jQuery的特点? Write Less, Do More

很好的解决了不同浏览器的兼容问题 (IE 6.0+, FF 2+, Safari 3.0+, Opera 9.0+, Chrome) css还是有问题的
对于不同控件具有统一的操作方式。

体积小 (几十KB)、**使用简单方便 (Write Less Do More)**
链式编程 `$("#div1").draggable().show().hide().fly()`、
隐式迭代

、**插件丰富、开源、免费**。插件多缺什么找什么
让编写JavaScript程序更简单、更强大!

学习js是打造装备,每天想着封装兼容,蛋疼

学习jquery:属性不够装备凑,蛋不疼,腰不酸,上楼都有劲了

John Resig

著有《Pro JavaScript Techniques》

（即《精通JavaScript》）

《Secrets of the JavaScript Ninja》

即《JavaScript忍者的秘密》

**John Resig has never made a carousel plugin..
when he goes to the carnival, ALL the rides support
jQuery.**

**John Resig can kill you with two lines of code. Sure,
he could do it with one, but the force of the
explosion would cause the universe to collapse
into a giant black hole.**



http://baike.baidu.com/link?url=zwDZ7ozCWR6zvCVJZGATLubjT4ntDgRvM_vZwE-bZ8iXGq5zhqE1mpr3HLLlEuiBiLRsAV4dvHaMWD64LvRuEa



```
$(function () {  
    $("#btn").click(function () {  
        $("#dv").css({ "width": "300px", "height": "200px", "backgroundColor": "red" });  
    });  
});
```

设置层的样式



```
$(document).ready(function () {  
    $("div").html("<p>这也可以</p>");  
});
```

这也可以

这也可以

这也可以

Jquery重点内容

重点:**Jquery使用**和**Jquery的选择器**

第一个难点:**Jquery对象和DOM对象互转**

基础目标:

Jquery基本使用

Jquery的常用选择器

案例

jQuery中的顶级对象\$

jQuery中最常用的对象即\$对象，要想使用jQuery的方法必须通过\$对象。只有将普通的Dom对象封装成jQuery对象，然后才能调用jQuery中的各种方法。

\$是jQuery简写，在代码中可以使用jQuery代替\$，但一般为了方便大家都直接使用\$。

写注释，后续jQuery代码会越写越多，所以必要的注释一定要写。

编写简单的jQuery代码

```
//Window.onload=function(){}页面中只能存在一个
window.onload=function () {
    console.log("页面完全加载后才触发该事件");
};
//$(document).ready(function(){});可以存在多个
$(docElem).ready(function () {
    console.log("1");
});
//$(document).ready(function(){});只要DOM元素加载完就触发
$(docElem).ready(function () {
    console.log("2");
});
```

```
//想在jquery中实现window.onload
$(window).load(function () {
    alert("哈哈1");
});
$(window).load(function () {
    alert("哈哈2");
});
//速度明显比window.onload快
$(document).ready(function () {
    alert("嘎嘎");
});
```

Window.onload只能写一次,重复会被后面的干掉
页面所有的标签,图片,外部引入加载后才触发
\$(document).ready()DOM基本标签加载后就触发
可以写多个
\$(document).ready()和\$(function)一样

```
$(function () {
    console.log("真的很帅");
});
$(document).ready(function () {
    console.log("哈哈,我又变帅了");
});
```

jQuery中引入文件注意问题

```
<script src="jquery-1.12.2.js"></script>
```

先引入文件,然后再使用
开发的时候引入正常的jquery文件和
压缩版的文件都没有问题,
建议:开发用普通版
上线用压缩版

为什么Jquery对象和DOM对象要

互转

DOM对象转Jquery对象操作方便,毕竟Jquery中方法都是封装好了的,而且兼容问题解决的很好,代码少,方便.

Jquery对象转DOM对象,因为Jquery中文件一直在更新,很多东西都是随着使用而进行封装和升级,不太可能把所有dom中用到的进行封装,还有很多未知的兼容问题没有封装进去,所以,有的时候jquery中不能解决的问题,还需要原生的js代码来解决,所以,jquery对象有的时候需要转成dom对象来进行操作

Jquery对象和DOM对象互转

```
$(function () {  
    var bt1=$("#btn")[0];//由jquery对象转成了dom对象  
    var bt2=document.getElementById("btn");//对象  
    console.log(bt1==bt2);  
});
```

如何将dom对象转成jquery对象,加\$就可以了

如何将jquery对象转dom对象

```
$(function () {  
    var dv1 = $("#dv");//由jquery对象转成了dom对象  
    var dv2 = document.getElementById("dv");//对象  
    //必然报错,为什么?因为jquery对象不能直接调用dom中对象的属性或者是方法  
    dv1.innerHTML = "<p>哈哈</p>";  
    //同理,下面这行代码必然报错,为什么呢?因为dom对象是不能直接调用jquery对象的方法  
    dv2.html("<p>哈哈</p>");  
});
```

`$("#btn")[0]`和`$("#btn").get(0)`都可以转dom对象, `$(dom对象)`即可转jquery对象

网页开关灯,鄙视DOM的写法

```
$(function () {  
    $("#btn").click(function () {  
        if($(this).val()=="开灯") {  
            $("body").css("backgroundColor", "black");  
            $(this).val("关灯");  
        }else{  
            $("body").css("backgroundColor", "");  
            $(this).val("开灯");  
        }  
    });  
});
```

```
$("#btn").click(function () {  
    if(this.value=="开灯") {  
        $("body")[0].style.backgroundColor="black";  
        this.value="关灯";  
    }else{  
        $("body")[0].style.backgroundColor="";  
        this.value="开灯";  
    }  
});  
});
```

JQuery选择器(重点)

选择器:就是为了获取元素的,
CSS中设置元素的样式可以通过选择器(c1-c3都用)
JQuery中通过选择器来获取元素进行操作
DOM中获取元素的方式:

`document.getElementById();`

`document.getElementsByTagName();`

`document.getElementsByClassName();`

代码多,长,麻烦,JQ中可以更简单,更方便

常用的JQ选择器,id选择器,标签选择器,类选择器

`$("#btn"),$("div"),$(".cls")`所有的

案例

1. 点击按钮, 设置层中的显示内容为: 这是一个层, 同时设置这个层的背景颜色(id选择器)
2. 点击按钮, 设置多个p标签中显示内容为: 我们都是p(标签选择器)
3. 点击按钮, 设置页面上应用cls类样式的元素的背景颜色(类选择器)

标签+类选择器

标签加类选择器

`$(li.cls)`

案例：

点击按钮设置页面上应用cls类样式li标签的背景颜色

多条件选择器

多条件选择器

`$(span,li,div)`

案例：

点击按钮,设置页面中的span标签,li标签,div标签的背景颜色

几个常见的方法

.html()方法,设置标签中间显示其他标签及内容,类似于innerHTML

.text()方法,设置标签中间显示的文本内容,类似于innerText

.val()方法.设置input标签中value的值,类似于value

.css()方法,.设置元素的样式,类似于style

注意:

()中写字符串,一个参数,就是要设置的值,什么也不写,返回的是这个属性的值

.css()写的是键值对

层级选择器

层次选择器:

后代选择器(子元素中元素,子,仔仔,孙)

`$("#dv li")`或者`$("#ul li")`或者`$(".cls li")`

子代选择器(直接的所有子元素,儿子)

`$("#ul>li")`或者`$("#div >span")`或者`$(".cls >li")`

案例:球队高亮显示

案例:点击按钮修改层中所有的标签的背景颜色

```
$(function () {  
    $("#u1 > li").mouseover(function () {  
  
    $(this).css("backgroundColor", "red");  
    });  
    $("#u1 > li").mouseout(function () {  
        $(this).css("backgroundColor", "");  
    });  
});
```

```
$("#dv li").css("backgroundColor", "red");  
  
:div id="dv">  
▼<ul id="u1">  
    ▼<li style="background-color: red;">  
        "公牛"  
        "  
    <ul>  
        <li style="background-color: red;">
```

层级选择器

获取当前元素的相邻元素:

`$("div + span")`

获取当前元素后面所有元素

`$("div ~ span")`

// 这个元素后紧跟着的第一个元素

```
$('div+span').css('backgroundColor', 'red');
```

// 这个元素后跟着的所有元素

```
$('div~span').css('backgroundColor', 'red');
```

案例: 下拉菜单

```
$(function () {  
    // 获取div下的ul下的li  
    $(".wrap>ul>li").mouseover(function () {  
        $(this).children("ul").show(); // 显示ul  
    });  
    $(".wrap>ul>li").mouseout(function () {  
        $(this).children("ul").hide(); // 隐藏ul  
    });  
});
```

帅哥系列

美女系列

渣男妇腐女

帅小杨

小杨帅

杨小帅

案例:隔行变色

乔峰:降龙十八摸

张无忌:乾坤大摸姨

段誉:鳞波微步

丁棚:天外流星贱

张三丰:太极掌

段飞:极乐神功

云飞扬:天饕功

杨过:黯然销魂掌

那谁谁:如来神掌

孟星魂:流星蝴蝶剑

```
$(function () {  
    alert("哈哈");  
    $("#u1  
li:odd").css("backgroundColor", "yellow");  
    $("#u1 li:even").css("backgroundColor", "blue");  
});
```

常见的选择器,(可以理解成是过滤器)

:even 偶数 选择器

:odd 奇数 选择器

索引选择器

eq(3)获取索引的元素

gt(3)索引大于3的所有元素

lt(3)索引小于3的所有的元素

```
$("#u1 li:eq(7)").css("backgroundColor", "red");  
$("#u1 li:gt(7)").css("backgroundColor", "green");  
$("#u1 li:lt(5)").css("backgroundColor", "black");
```



段飞:极乐神功

云飞扬:天饕功

杨过:黯然销魂掌

那谁谁:如来神掌

孟星魂:流星蝴蝶剑

案例:淘宝精品展示



```
$(function () {  
    $("#left > li").mouseover(function () {  
        var index=$(this).index();  
        //第一种方式  
        $("#center > li").hide();  
        $("#center > li:eq("+index+"").show();  
        //第二种方式  
        //$("#center > li:eq("+index+"").siblings().hide();  
        //$("#center > li:eq("+index+"").show();  
        //第三种方式  
        //$("#center > li:eq("+index+"").show().siblings().hide();  
    });  
    $("#right > li").mouseout(function () {  
        var index=$(this).index();  
        $("#center > li").hide();  
        $("#center > li:eq("+index+9+"").show();  
    });  
});
```

案例:突出显示图片

```
$(".wrap li").mouseover(function () {  
    $(this).css("opacity",1).siblings("li").css("opacity",0.5);  
});  
$(".wrap").mouseout(function () {  
    $(this).find("li").css("opacity",1);  
});
```



.find()查找元素

案例:手风琴

```
$("#box ul > li").mouseover(function () {  
    $(this).css("width", "800px").siblings("li").css("width", "100px");  
});  
$("#box ul").mouseout(function () {  
    $("li").css("width", "240px");  
});
```

```
$('#u1 li ul li').hide();  
$('#u1 li').click(function () {  
    $('#u1 li', $(this)).show(500);  
    $('#u1 li', $(this).siblings('li')).hide(500);  
});
```

幼儿园同学

鼻涕虫

爱哭鬼

张大胆

小学同学

初中同学

```
$(".groupTitle").click(function () {  
    $(this).next("div").show();  
    $(this).parent("li").siblings("li").children("div")  
});
```

酒水

饮料

茶叶

龙井大红袍

Jquery操作样式

//第一种写法:

```
$("#u1>li").css("backgroundColor","red");  
$("#u1>li").css("fontSize","全新硬笔行书简");  
$("#u1>li").css("fontSize","30px");
```

//第二种写法:

```
var style={  
    "backgroundColor":"red",  
    "fontSize":"全新硬笔行书简",  
    "fontSize":"30px"  
};
```

```
$("#u1>li").css(style);
```

//第三种写法:

```
$("#u1>li").css("backgroundColor","red").css("fontSize","全新硬笔行书简")  
).css("fontSize","30px");
```

```
console.log($("#u1>li").css("backgroundColor"));
```

获取属性的值,写一个参数即可

Jquery操作类样式

```
$("#btn1").click(function () {  
    $("#dv").addClass("cls1");//添加一个样式  
    $("#dv").addClass("cls1 cls2");//添加多个样式  
    $("#dv").addClass("cls1").addClass("cls2");  
});  
$("#btn2").click(function () {  
    //移除某个样式  
    $("#dv").removeClass("cls1");  
    //移除多个样式  
    $("#dv").removeClass();  
});
```

addClass();参数:类样式名字，添加样式的同时不会干掉原有的样式

removeClass();不写参数移除所有的类样式

removeClass("cls");移除指定的一个类样式

//判断该元素是否有这个类样式

```
console.log($("#dv").hasClass("cls2"));
```

```
$("#body").toggleClass("cls");
```

案例:实现开关灯

```
$("#btn").click(function () {  
    if($("#body").hasClass("cls"))  
        $("#body").removeClass("cls");  
    $(this).val("开灯");  
} else {  
    $("#body").addClass("cls");  
    $(this).val("关灯");  
}  
});
```

一行代码实现开关灯

案例:tab键切换产品内容

```
$(".tab>li").mouseenter(function () {  
    $(this).addClass("active").siblings("li").removeClass("active");  
    // 第一种写法: 索引选择器eq  
    $(".products>div:eq("+$(this).index()+")").addClass("selected").siblings("div").removeClass("selected");  
    // 第二种写法: 索引方法eq(index)  
    $(".products>div").eq($(this).index()).addClass("selected").siblings("div").removeClass("selected");  
});
```

国际大牌 | 国妆名牌 | 清洁用品 | 男士精品



Chanel香奈儿香水女士发香喷



伊夫圣罗兰 YSL 美丽鸦片女士

css和类样式对比

设置的样式如果很多那么直接使用类样式的方式进行操作

为了方便维护也最好使用类样式进行操作
简单的操作直接使用css的方式

链式编程:好处大大地!

多行代码合并成一行代码,前提要认清此行代码返回的是不是对象.是对象才能进行链式编程

.html('val').text('val').css()链式编程，隐式迭代。

链式编程注意：\$('div').html('设置值').val('设置值');这样可以，但是\$('div').html().text()这样是不对的，因为获取值时返回的是获取的字符串而不是对象本身所以不能链式编程。

【通过开发人员工具调试方法。】

```
$("#u1>li").css("backgroundColor", "red").css("fontSize", "全新硬笔行书简").css("fontSize", "30px");
```

```
$('#p').text('我们都是p').css('border', '1px solid red').click(function () {  
    alert($(this).text());  
});
```

获得兄弟元素的几个方法

`next();` //当前元素之后的紧邻着的第一个兄弟元素（下一个）

`nextAll();` //当前元素之后的所有兄弟元素

`prev();` //当前元素之前的紧邻着的兄弟元素（上一个）

`prevAll();` //当前元素之前的所有兄弟元素

`siblings();` //当前元素的所有兄弟元素

案例：页面上有一个ul球队列表当鼠标移动到某个li上的时候改行背景颜色变红，当点击某个li的时候，让该li之前的所有li背景色变黄，之后的所有li背景色变蓝。自己不变色。注意：`nextAll()`、`prevAll()`等方法返回值是一个元素集合，这里链式编程时要想清楚当前返回的值是什么。

```
$("#ul>li").mouseenter(function () {  
    $(this).css("backgroundColor", "red").siblings().css("backgroundColor", "");  
}).click(function () {  
    $(this).prevAll().css("backgroundColor", "yellow");  
    $(this).nextAll().css("backgroundColor", "blue");  
}).mouseleave(function () {  
    $(this).css("backgroundColor", "");  
});
```

当化 (SNOW)

奥丁格教士
(Franziskaner)

科罗娜喜力柏龙
(Paulaner)

嘉士伯Kaiserdom

罗斯福 (Rochefort)



jQuery动画

hide()方法:

show()方法:

参数:可以是一个number类型,也可以是字符串类型

```
$("#div").hide(1000); // 参数是数字
$("#div").hide("slow"); // 参数是字符串200ms
$("#div").hide("normal"); // 参数是字符串400ms
$("#div").hide("fast"); // 参数是字符串600ms

$("#div").hide(1000, function () {
    alert("隐藏完毕");
});
```

```
$("#div").show(1000); // 参数是数字
$("#div").show("slow"); // 参数是字符串200ms
$("#div").show("normal"); // 参数是字符串400ms
$("#div").show("fast"); // 参数是字符串600ms
$("#div").show(1000, function () {
    alert("显示完毕");
});
```

隐藏动画案例

```
$("#btn1").click(function () {  
    $("div>img:last-child").hide(1000,function () {  
        //使用arguments.callee所以我们不需要指定函数  
        $(this).prev().hide(1000,arguments.callee);  
    });  
});  
$("#btn2").click(function () {  
    $("div>img:first-child").show(1000,function () {  
        //使用arguments.callee所以我们不需要指定函数  
        $(this).next().show(1000,arguments.callee);  
    });  
});
```

隐藏动画

显示动画



```
$("#div").click(function() {  
    $( this ).hide( 2000, function() {  
        $( this ).remove();  
    });  
});
```



其他动画方法

```
$("#div").slideUp(1000); // 滑入  
$("#div").slideDown(1000); // 滑出  
$("#div").slideToggle(1000); // 切换滑入滑出  
$("#div").fadeIn(1000); // 淡入  
$("#div").fadeOut(1000); // 淡出  
$("#div").fadeToggle(1000); // 切换淡入淡出
```

```
// 可以设置透明度, 参数1: 时间, 参数2: 到达透明度  
$( "div" ).fadeTo(1000, 0.3); // 一个参数, 常用  
// 参数1: 没有时间, 参数2: 到达透明度  
// 和css("opacity", 0.1);  
$("#div").fadeTo(0, 0.1); // 一个参数
```

动画方法:animate方法

```
$("#im").animate({"left": "10px", "top": "500px", "width": "50px", "height": "50px"}, 2000, function () {  
    $("#im").animate({"left": "+=505px", "top": "-=400px", "width": "+=200px", "height": "+=200px"}, 1000);  
});
```

第一个参数:键值对,(数值的属性可以改,颜色不能改)

第二个参数:动画的时间

第三个参数:回调函数

停止动画效果:stop()方法,解决下拉框案例中的bug

```
$(".wrap>ul>li").mouseover(function () {  
    $(this).children("ul").stop().show(500); //显示ul  
});  
$(".wrap>ul>li").mouseout(function () {  
    $(this).children("ul").stop().hide(500); //显示ul  
});
```

jQuery中创建元素及追加元素

DOM中可以动态创建元素:document.createElement(“标签的名字”);

jQuery中同样可以创建元素标签,并且返回的就是jQuery对象,可以直接调用方法进行使用

append方法用来在元素的末尾追加元素（最后一个子节点）。增加元素末尾(儿子)

prepend，在元素的开始添加元素（第一个子节点）。增加元素开始(儿子)

after，在元素之后添加元素（添加兄弟）增加元素后面(兄弟)

before：在元素之前添加元素（添加兄弟）增加元素前面(兄弟)

```
$('#dv').append($('
```

.append方法可以把页面中的元素添加到其他标签中,添加的方式属于**移动**

```
$("#div").append($("#p"));
```

案例:权限选择



```
$("#toRight").click(function () {  
    $("#se1 option:selected").appendTo($("#se2"));  
});  
$("#toLeft").click(function () {  
    $("#se2 option:selected").appendTo($("#se1"));  
});  
$("#toAllLeft").click(function () {  
    $("#se2").append($("#se1 option"));  
});  
$("#toAllRight").click(function () {  
    $("#se1").append($("#se2 option"));  
});
```


.html方法

```
// 设置内容后, 原内容丢失, 返回的是当前对象  
// var dvObj=$("#div").html("哈哈");  
// 设置层中间显示其他的标签内容  
$("#div").html("<a href='http://www.baidu.com'>百度</a>");  
// 直接获取的是标签中的内容  
console.log($("#div").html());
```

.html方法设置内容,返回的是当前的对象,如果不传入参数,获取的是标签中的内容

.text方法同上

创建元素的方式:

.html方法可以创建元素

\$("#html标签")可以创建元素

案例:点击按钮动态创建表格

```
for(var i=0;i<datas.length;i++){  
    var dt=datas[i];  
    $("<tr><td><a href='"+dt.url+"'>"+dt.name+"</a></td><td>"+dt.type+"</td></tr>").appendTo($("#tbd"));  
}
```

```
var array=[];  
for(var i=0;i<datas.length;i++){  
    var dt=datas[i];  
    array.push("<tr><td><a href='"+dt.url+"'>"+dt.name+"</a></td><td>"+dt.type+"</td></tr>");  
}  
$("#tbd").html(array);
```

两种做法都可以

案例:清空层中的内容和移除层

```
$("#div").html("");//清空层中内容  
$("#div").empty();//清空层中的内容  
$("#div").remove();//干掉自己
```

```
$("#btn").click(function () {  
    var cloneUl=$("#dv1").children().clone();  
    cloneUl.css("fontSize","50px");  
    $("#dv2").append(cloneUl);  
});
```

clone方法,克隆标签内容

名称	说明
传智播客	IT最强培训机构
黑马程序员	大学生IT培训机构
传智前端学院	前端黄埔军校

设置和获取表单的value

input标签:文本框,radio,select,textarea(文本域),checkbox....常见的表单标签
都可以通过val方法获取和设置value值

```
$("#txt1").val("admin");  
$("#txt2").val("123456");  
$("#txt3").val("这是一个条款");  
$("#s1").val(2); // 设置value为2的选中
```

```
console.log($("#txt1").val());  
console.log($("#txt2").val());  
console.log($("#txt3").val());  
console.log($("#s1").val());
```

案例:添加课程

添加数据

课程名称	所属学院	已学会
JavaScript	传智播客-前端与移动开发学院	GET
css	传智播客-前端与移动开发学院	GET
html	传智播客-前端与移动开发学院	GET
jQuery	传智播客-前端与移动开发学院	GET

案例:设置和获取系统属性值或者自定义属性

```
$("#btn1").click(function () {  
    $("a").attr("title", "传智播客")  
    $("a").attr("href", "http://www.itcast.cn");  
});  
$("#btn2").click(function () {  
    // 获取属性值  
    console.log($("a").attr("title"));  
});
```

attr()方法:可以设置属性值,两个参数:1.属性名字,2.属性值

attr()方法:可以获取属性值,一个参数:1.属性名字

案例:全选和反选案例

案例:设置复选框选中:(attr设置checkbox的选中有问题)prop
prop用法和attr一样

```
var checked = $("#cbx").attr("checked");
console.log(checked);
if(checked == undefined) {
    console.log("没选中");
    $("#cbx").attr("checked", true);
} else {
    console.log("已经选中");
    $("#cbx").attr("checked", false);
}
```

```
$("#j_cbAll").click(function () {
    var checked=$("#this").prop("checked");
    if(checked){
        $("#j_tb").find(":checkbox").prop("checked",true);
    }else{
        $("#j_tb").find(":checkbox").prop("checked",false);
    }
});
$("#j_tb").find(":checkbox").click(function () {
    //查找这里所有的checkbox的个数
    var checkboxLength=$("#j_tb").find(":checkbox").length;
    //查找选中的checkbox的个数
    var checkboxChecked=$("#j_tb").find(":checked").length;
    if(checkboxLength==checkboxChecked){
        $("#j_cbAll").prop("checked",true);
    }else{
        $("#j_cbAll").prop("checked",false);
    }
});
```

```
var checked = $("#cbx").prop("checked");
if (!checked) { //没选中
    $("#cbx").prop("checked", true);
} else { //选中
    $("#cbx").prop("checked", false);
}
```

<input checked="" type="checkbox"/>	课程名称	所属学院
<input checked="" type="checkbox"/>	JavaScript	前端与移动开发学院
<input checked="" type="checkbox"/>	css	前端与移动开发学院
<input checked="" type="checkbox"/>	html	前端与移动开发学院
<input checked="" type="checkbox"/>	jQuery	前端与移动开发学院

设置和获取元素的宽和高

```
$("#btn1").click(function () {  
    // 字符串类型  
    console.log($("#div").css("width"));  
    console.log($("#div").css("height"));  
});  
$("#btn2").click(function () {  
    var width=parseInt($("#div").css("width"))*2;  
    var height=parseInt($("#div").css("width"))*2;  
    $("#div").css("width",width+"px");  
    $("#div").css("height",height+"px");  
});
```

通过.css()方法写一个属性可以获取宽或者高,是字符串类型
如果获取后重新设置需要转换,麻烦

```
console.log($("#div").width());  
console.log($("#div").height());
```

获取的时候直接就是数字类型

```
// 直接设置数字的值即可  
$("#div").width(500);  
$("#div").height("400px");
```

位置操作

```
// 获取的是对象, 并且没有px  
console.log($("#div").offset());  
console.log($("#div").offset().left);  
console.log($("#div").offset().top);
```

```
// 设置的时候可以没有px,  
// 设置前如果没有脱离文档流, 设置的时候会有relative  
// 设置前如果有脱离文档流, 设置的时候不会再次设置脱标  
$("#div").offset({left:100,top:100});
```

Offset()方法返回的是对象,并且,对象中有一个left和一个top,并且值是数字类型

设置的时候也可以没有px

设置的时候元素在设置前如果没有脱离文档流,设置的时候会自动进行脱离文档流,默认为relative

如果设置前有脱离文档流,那么设置的时候直接改变位置

注意:如果层和层中的标签(p)同时定位,并且,p距离左上角50px.通过按钮设置层距离左上角100px,那么p此时距离左上角为150px

scrollLeft()和scrollTop

```
console.log($(document).scrollLeft());  
console.log($(document).scrollTop());
```

用法和之前的offset()方法类似
获取的是数字
设置可以在括号中添加值即可

```
$(function () {  
    $(window).scroll(function () {  
        var h=$(".top").height();  
        var top=$(window).scrollTop();  
        if(top>=h){  
            $(".nav").css({"position":"fixed","top":0});  
            //此时main为固定定位，所以设置margin-top  
            $(".main").css({"marginTop":$(".nav").height()});  
        }else{  
            $(".nav").css({"position":"static"});  
            //此时main为固定定位，所以设置margin-top  
            $(".main").css({"marginTop":0});  
        }  
    });  
});
```

案例:固定导航栏

\$(function)文档加载后执行
所以有可能获取不到图片高度

解决:\$(window).load();
一种在style标签中设置样式

事件绑定

// 同一个元素绑定多个事件麻烦

```
$("#btn").click(function () {  
    alert("被点了");  
});  
$("#btn").mouseover(function () {  
    $(this).css("backgroundColor", "red");  
});  
$("#btn").bind("click", function () {  
    alert("被点了");  
});
```

绑定事件: **.bind** 方法

方便: 简单, 多个事件一次绑定
参数: 键值对(右), 空格隔开(下)

```
var i=0; // 多个事件处理的问题相同还可以这么玩  
$("#btn").bind("mouseover mouseout", function  
( ) {  
    i++;  
    console.log(i);  
});
```

```
$("#btn").bind({ "click": function ( ) {  
    alert("哎呀妈呀");  
}, "mouseover": function ( ) {  
    $(this).css("backgroundColor", "red");  
}, "mouseout": function ( ) {  
    $(this).css("backgroundColor", "");  
}});
```

bind方法可以为动态创建元素绑定事件,但是不怎么用

```
$("#btn").bind("click",function () {  
    $("div").append($("<p>创建后的p</p>"));  
    //参数1:要触发事件的动态元素  
    //参数2:要触发的事件名字(事件的类型)  
    //参数3:执行事件的函数  
    $("div").delegate("p","click",function () {  
        alert("哎呀");  
    });  
});
```

delegate()方法为父级元素绑定事件,子元素来触发.

绑定事件on

delegate()方法内部也是调用的on方法来绑定事件的

```
$("#btn").on("click",function () {  
    $("#div").append($("#<p>创建后的p</p>"));  
    $("#div").first().on("click",function () {  
        alert("哎呀");  
    });  
});
```

```
/*  
* 参数1: 绑定事件的类型  
* 参数2: 绑定事件的元素  
* 参数3: 执行事件的函数  
* */  
$("#div").on("click","p",function () {  
    alert("哎呀");  
});
```

<http://www.css88.com/doc/zeptojs/#on> 手机开发中也是使用on的方式绑定事件



解决动态创建表格添加元素绑定事件bug

```
$("#j_tb").on("click", ".get", function () {  
    $(this).parent("td").parent("tr").remove();  
});
```

从此以后绑定事件用这个会更好,更完美,更好玩,很现代的方式
节约内存,效率高,时间少

解绑事件

.off()方法可以解绑事件

```
$("#btn1").on("click",function () {  
    alert("被点了");  
});  
$("#btn2").on("click",function () {  
    $("#btn1").off("click");  
});
```

推荐:用什么方式绑定事件就要用对应的解绑事件解除绑定

```
$("#btn").bind("click",function(){});  
$("#btn").unbind("click");  
$("#dv").delegate("p","click",function(){});  
$("#dv").undelegate("p","click");
```

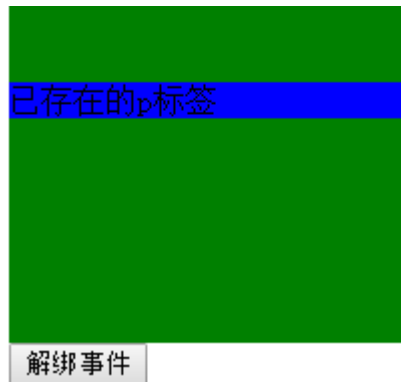
//解绑多个事件

```
$("#btn1").off("click mouseover");
```

多个事件解绑 加 空格

解绑事件

```
$("#div").on("click",function () {  
    alert("层被点了");  
});  
$("#div").on("click","p",function () {  
    alert("p被点了");  
});  
$("#btn2").on("click",function () {  
    //解绑事件的时候div和p同时解绑(所有的)  
    //自身绑定的事件以及动态绑定的事件都会解绑  
    $("#div").off("click");  
});
```



解除绑定事件:自身的绑定事件和动态绑定的事件都会被解绑

```
$("#div").off("click","**");
```

```
$("#div").off();
```

全部解绑

固定的写法,解除p的绑定事件,自身div的事件不会被解绑

触发事件

触发事件:触发某个事件的时候在该事件内部调用了其他元素的某个事件方法

第一种方式:直接调用元素的事件方法:

```
$("#div").click();
```

第二种方式:使用.trigger()方法,扳机

```
$("#div").trigger("click");
```

第三种方式:使用.triggerHandler()方法

```
$("#div").triggerHandler("click");
```

```
$("#txt").on("focus",function () {  
    $(this).next().text("获取了焦点");  
});  
$("#btn").on("click",function () {  
    $("#txt").trigger("focus");  
    $("#txt").triggerHandler("focus");  
});
```

.trigger()和.triggerHandler()区别

前者会触发浏览器的默认行为,并执行事件
后者不会触发浏览器默认行为,但是会执行事件

获得焦点属于浏览器的默认行为

事件对象

```
$("#div").on("click",function (event) {  
    console.log(event);  
});
```

```
▶ event.currentTarget: div  
▶ event.delegateTarget: div  
▶ event.target: div
```

触发事件

event.delegateTarget:代码绑定事件的对象

event.currentTarget:绑定事件的对象

event.target:真正触发事件的对象

```
$("#div").on("click","input",function (event) {  
    console.log(event);  
});
```

```
▶ event.currentTarget: input#btn  
▶ event.delegateTarget: div  
▶ event.target: input#btn
```

```
$(document).on("keydown",function (event) {  
    console.log(event.keyCode);  
});
```

```
event: jQuery.Event  
  altKey: false  
  bubbles: true  
  button: 0  
  buttons: 0  
  cancelable: true  
  clientX: 253  
  clientY: 106  
  ctrlKey: true  
▶ currentTarget: div  
  data: undefined  
▶ delegateTarget: div  
  detail: 1  
  eventPhase: 2  
  fromElement: null  
▶ handleObj: Object
```

按键的键值网页中经常使用

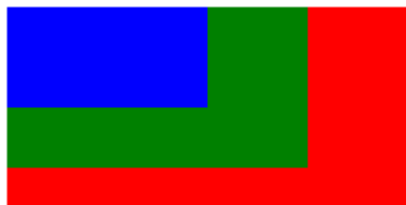
案例：

```
$(document).on("keydown",function (event) {  
    $("#keyCodeSpan").text(event.keyCode);  
    switch (parseInt(event.keyCode/10)){  
        case 6: changeColor("red");break;  
        case 7: changeColor("green");break;  
        case 8: changeColor("blue");break;  
    }  
});  
function changeColor(color) {  
    $("#bgChange").css("backgroundColor",color);  
}
```

按键改变颜色

keyCode为: 18

取消事件冒泡+取消默认事件



取消事件冒泡:
e.stopPropagation()
或者
return false;

```
$("#dv1").on("click",function () {  
    alert("第一个层");  
});  
$("#dv2").on("click",function () {  
    alert("第二个层");  
});  
$("#dv3").on("click",function (event) {  
    alert("第三个层");  
    //取消事件冒泡  
    event.stopPropagation();  
    return false;也可以的  
});
```

```
$("#a").on("click",function () {  
    alert("超链接");  
    return false;  
});
```

return false

不仅可以用来取消事件冒泡
同样可以取消默认事件

```
event.preventDefault();
```

也可以取消默认事件

链式编程原理

```
function Student() {  
    this.name="小明";  
    this.sayHi=function (sayText) {  
        console.log(this.name+"说:"+sayText);  
        return this;  
    };  
    this.addMoney=function (money) {  
        console.log(this.name+"说"+money+"不给我");  
        return this;  
    };  
}  
var stu=new Student();  
stu.sayHi("干啥去").addMoney(100);
```

链式编程原理:内部返回了return this当前对象

有些方法设置了值才能返回当前对象，如果没有设置值，是获取属性对应的值,而不是当前对象

```
this.sayHello=function (txt) {  
    if(txt){  
        return this;  
    }else{  
        return "你说啥了";  
    }  
};
```

案例:评分

```
$(".comment>li").on("mouseover",function () {  
    $(this).text("★").prevAll("li").text("★").end().nextAll("li").text("☆");  
}).on("mouseout",function () {  
    $(this).parent().children("li").text("☆");  
    //第二步:2  
  
$(".comment>li[index=1]").text("★").prevAll("li").text("★").end().nextAll("li").text("☆");  
}).on("click",function () {  
    //第二步:1  
    $(this).attr("index",1).siblings("li").removeAttr("index");  
});
```



each方法

jQuery中有隐式迭代,不需要我们再次进行遍历对某些元素进行操作
但是,如果涉及到对不同的元素有不同的操作,那么需要进行each遍历

```
// 参数: 表示每次遍历都要执行的函数
$("li").each(function (index, ele) {
    // index 表示: 当前这个元素的索引号 从0开始的 0 - 10
    var op = (index + 1) / 10;
    $(ele).css("opacity", op);
});
```



多库共存

```
// 让jQuery释放对$的控制权
$.noConflict();
var $="哦买噶";
console.log($);
jQuery(function () {
    jQuery("#btn").click(function () {
        alert("正常的执行");
    });
});
```

同一个页面不仅引入了jQuery的外部文件,也引入了其他的库文件
如果此时其他的库文件中也使用了\$符号
那么就使用:\$.noConflict()解决
其他语言中:这个方式叫解决命名空间的冲突
也可以这样:

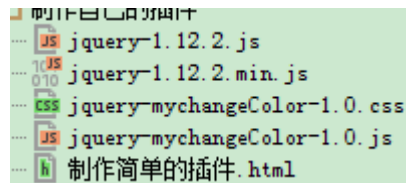
```
var itcast=$.noConflict();
var $="哦买噶";
console.log($);
itcast(function () {
    itcast("#btn").click(function () {
        alert("正常的执行");
    });
});
```

jQuery插件玩法

- 1.引入jquery文件
- 2.引入插件的js文件(还需要引入css文件)
- 3.添加加载的事件
- 4.复制代码即可

```
$("div").animate({  
    "background-color": 'green'  
}, 2000);
```

自己做插件



```
<script src="jquery-1.12.2.js"></script>
<script src="jquery-mychangeColor-1.0.js"></script>
<link rel="stylesheet" href="jquery-mychangeColor-1.0.css"/>
```

```
$(".cls").changeBackgroundColor($(".this").val()).text("嘎嘎");
```

在线桌面帮助文档

<http://www.jq-school.com/api/index.html>

jqueryUI使用步骤

1. 引入jQueryUI的样式文件
2. 引入jQuery
3. 引入jQueryUI的js文件
4. 使用jQueryUI功能

jQuery的迭代(包装集)

- 如何判断对象是否存在，jQuery选择器返回的是一个对象数组，调用text()、html()、click()之类方法的时候其实是对数组中每个元素迭代调用每个方法，因此即使通过id选择的元素不存在也不会报错，如果需要判断指定的id是否存在，应该写：
- ```
if ($("#btn1").length <= 0) {
```
- ```
    alert("id为btn1的元素不存在！");
```
- ```
}
```
- `$("#id").length>0`//用途：可以判断页面上的某个元素是否存在，如果存在则不再重新创建，如果不存在则创建一个新的并显示（动态创建元素的时候用。）

[http://www.w3school.com.cn/tiy/t.asp?f=jquery\\_event\\_mouseenter\\_mouseover](http://www.w3school.com.cn/tiy/t.asp?f=jquery_event_mouseenter_mouseover)



# Thank You!

改变中国 IT 教育，我们正在行动

www.itcast.cn