# jQuery

## 基础

语法：**$(*selector*).*action*()**

选择器：$()开头，基于元素的 id、类、类型、属性、属性值等"查找"（或选择）HTML 元素。

例：**$("#test")** **$(".test") $("\*") $(this)**(当前元素) **$("p.intro")**(类为intro的p元素) $("p:first")(第一个p元素) $("ul li:first")(选取第一个ul的第一个li元素)

$("ul li:first-child")(所有ul的第一个li) **$("[href]")(带有href的属性)**

$("a[target!='\_blank']") **(选取所有非target属性值的a元素)** $("tr:even") **(偶数位置的tr)**

常见事件：



**change方法**：当元素的值发生改变时，会发生 change 事件，change() 函数触发 change 事件，或规定当发生 change 事件时运行的函数；该事件仅适用于文本域（text field），以及 textarea 和 select 元素。

可用于全选框制作

vue()方法：

attr和prop的区别，prop用在CheckBox

## 效果

### 隐藏与显示 hide() & show()

$("#demo").click(function(){ $("p").hide(); }); 效果同css的display: none

$("#demo").click(function(){ $("p").show(); });效果同css的display: block

hide(speed,callback); 可以规定隐藏的速度，callback 参数是隐藏或显示完成后所执行的函数名称

例：$("#demo").click(function(){ $("p").show(1000, function(){alert(“文本已隐藏”)}); });

### 淡入淡出 fadeIn() fadeOut() fadeToggle()

淡入：即将隐藏的元素显现出来，可设置显示的速度slow,fast或者毫秒

$(*selector*).fadeIn(*speed,callback*);

淡出：将显示的元素隐藏起来，$(*selector*).fadeOut(*speed,callback*);

自动判断：若元素是显现则隐藏，若元素隐藏则显现

$(selector).fadeToggle(speed,callback);

淡入淡出程度：必须要有slow/fast/Time ，可以规定隐藏的程度0.5

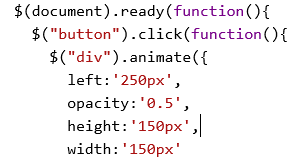
$(selector).fadeTo(speed,opacity,callback

### 滑动 slideDown() slideUp() slideToggle()

一般用于button和hover时下拉出菜单栏或者内容，使用方法同上

### 动画 animate()

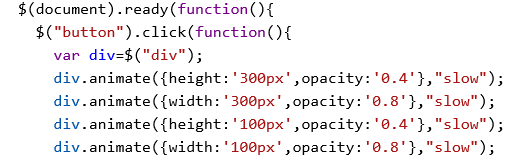
1. 可以操控多个属性，可以操控几乎所有css的属性



1. 可用相对值：height: +=250px
2. 设定动画属性：height: toggle show hide

点击按钮图片从下到上消失或者从上到下显现

1. 同时使用多个动画



将会一个一个显示

1. .stop()可以设置按钮停止动画 .stopAll(true) 停止所有动画
2. 每个动画结束会触发回调函数，停止的动画不会触发

### 方法链接

$("#p1").css("color","red").slideUp(2000).slideDown(2000);

将多个方法串连，先将颜色变红，然后滑动移出，最后滑动显现

## HTML

### 获取和设置

1. 获得内容：$(“#id”).text() .html()  **.val()** （表单字段）
2. 获得属性：$(“#id”).attr(“href”) **prop(“href”)**

一般固有属性(非自己定义的)用prop

1. 设置内容和属性：$(“#id”).text(“hello world”) 往里面加东西就是了，可以添加函数，处理字段

### 添加与删除

1. 添加新元素 append() pappend() after() before()

例：$("p").append("追加文本");

例：$("p").pappend(“在最前面追加文本");

例： $("img").after(“在图片后追加文本");

例： $("img").before(“在图片前追加文本");

$("img").after(txt1,txt2,txt3); 可一次添加多个文本

1. 删除元素 remove() empty()

$("p").remove(); $("p").remove(".italic");

$("#id").empty(); 删除子元素

$("p").remove(".italic"); 删除class=italic的p元素

1. 添加标签 添加一个input并获取焦点

$('.box').click(function(){  
**$('<input type="text"/>').appendTo('.box').focus();**  
});

var inp = **$('<input type="text"/>');**

$("p").pappend(inp); 添加新元素到p元素前

### CSS

1. 添加类：$(“#id”).addClass(“.className”)
2. 移除类：removeClass(“.className”) 可删除多个类
3. 切换：toggleClass(“.className”) 有则删除，无则加之
4. CSS：设置获取css的样式属性

获取：$(“#id”).css(“background”) $(“#id”).css(“width”)

设置：$(“#id”).css(“background”, “1px solid blue”)

设置多个：.css({“background”: “1px solid blue”, “color”: “white”})

方法：

$("#div1").width() $("#div1").height()

$("#div1").innerWidth()内边距+content $("#div1").innerHeight()

$("#div1").outerWidth() 边框+内边距+content

### 遍历

parent() parents() parentsUntil()

children() find()

同级：siblings() next() nextAll() nextUntil() prev() prevAll() prevUntil()

过滤：first() last() eq(1) filer() not()

$("span").parents().css() 把所有span的样式进行设定

$("span").parentsUntil(“#id2”).css() span到id为id2时的父元素

**$("id2 ").find(“span”).css()** 一般是在子节点的下层

**$("id2 ").siblings()**.css() 所有同级

**$("id2 ").next()**.css() id2的下一个同级元素 nextAll()下面所有同级元素

$(“id1”).nextUntil(“id4”).css() 从id1到id4之间

prev() 往前查找

$(“id1 p”).first().css() id1里面第一个p元素

$(“id1 p”).eq(1).css() id1里面第二个p元素

$(“div”).filer(“.className”) class=”className”的div块

$(“div”).not(“.className”).css() 除了class=”className”的div块

not加上eq可以实现反选 not(“:eq(4)”)

### AJAX

**AJAX 是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术**

ajax = 异步js + xml

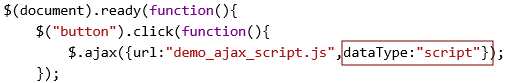
1. **ajax()方法**

语法：$.ajax({name:value, name:value, ... })

用于执行ajax的（http异步）请求

请求参数：async 布尔值，表示请求是否异步，默认true







1. **load()** 方法从服务器加载数据，并把返回的数据放入被选元素中。

语法：$(selector).load(URL,data,callback);

例：$("#div1").load("demo\_test.txt #p1");

可已将txt文件中的id=p1的部分加载到id=div1的模块中，会覆盖原文本内容

**实战用法：**

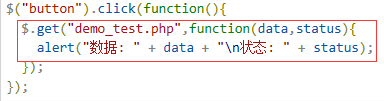
为了避免多页面情形下的代码重复，可以利用 load() 方法，将重复的部分（例如导航栏）放入单独的文件，使用下列方法进行导入js文件：

**load()**方法通常用来从Web服务器上获取静态的数据文件，然后这并不能体现Ajax的全部价值。在项目中，如果需要传递一些参数给服务器中的页面，那么可以使用**$.get()**或者**$.post()**或者**$.ajax()**方法

1. **$.get()、$.post()**

get方法通过http get请求向服务器发送请求数据

**语法**：$.get(*URL*,*callback*);



$.post() 方法通过 HTTP POST 请求向服务器提交数据。

语法：$.post(*URL,data,callback*);



1. load、get、post方法是对$.ajax()方法的封装，如需使用更为复杂的请求就需要用到$.ajax()方法

# 补充方法

1. val() 用于获取和设置匹配到的元素的值

$("input").keyup(function(){

var a = $(this).val() 获取在input中输入的值

$("#text").val(a + "Hello World"); 自定义输出

});

原生js中可用：

document.getElementById().value

若要使用类名：

document.getElementsByClassName('input')[0].value

1. change() 被选元素的值和状态发生改变，触发change事件，原生js不好实现。
2. prop() 用于获取和设置属性值

$('thead  input').change(**function** () {

**var** a = $(this).prop('checked');

       $('tbody input').prop('checked',a)

});

注：thead中的input定义为全选按钮，tbody中的为单个复选按钮；prop获取全选按钮的checked属性的状态，并将该属性值设置给其他单个复选按钮；当全选状态改变时，change事件触发，其他按钮一同改变。

该方案通过原生js的onchange()和addEventListener('change', function (){})

方法只能触发一次，多次点击无效。只能通过两个按钮控制checked = true或者false.

