## jQuery

js库有哪些：Prototype、YUI(网络反响一般)、Dojo、ExtJS、jQuery

什么是jQuery：就是js库中的一种,封装了很多的js的代码(函数)

jQuery的优点:用更少的代码完成更多的功能 事半功倍 免费,开源,插件多,链式编程,隐式迭代

BOM中的顶级对象: Window

DOM中的顶级对象: Document

jQuery中顶级对象:jQuery,或者是$

$和jQuery 可以互相代替

## 页面加载事件的不同方式

方法一 页面所有内容全部加载完后才可执行

$(window).load(function(){

});

方法二 页面中基本的标签加载后就可执行

$(document).ready(function(){

});

方法三 页面中基本的标签加载后就可执行

jQuery(function(){

});

方法四 页面中基本的标签加载后就可执行

$(function(){

});

注：方法一 比 方法二三四 都晚执行

DOM的页面加载方式：

window.onload = function () {

}；

## jQuery对象和DOM对象互转的问题

### jQuery对象转DOM对象两种方式:

第一种: jQuery对象[0];

第二种: jQuery对象.get(0);

### DOM对象转jQuery对象

var btnObj2=$(btnObj);

this---DOM对象

$(this)---jQuery对象

## 选择器

id选择器

$(‘#’)

标签选择器

$(‘ ’)

类选择器

$(‘. ’)

并集选择器

$(‘ , , , , ’)

交集选择器

$(‘div.cls’) div下的cls类

其他选择器

$(‘ div p’) 空格表示div中所有的p（包括儿子孙子 只要是div中的就行）

$(‘ div>p’) > 表示div子代中的p 隔代的不选 只选直接子元素

$(‘ div+p’) > 表示div紧邻的那个p

$(‘ div~p’) ~ 表示div 后面的兄弟元素 不一定要挨着

$(‘li[xiabiao=1]’) 根据（自定义）属性选择

过滤选择器

:odd 意思是奇数 实际是偶数 因为下标从0开始

:even 意思是偶数 实际是奇数

$(' #uu : odd ')

：eq（）

索引值为..的元素 索引值从0开始算

:selected

选中的

### 筛选选择器

..中的子元标签

(‘$父标签’)．children(子标签)

..的兄弟标签中的..标签

(‘$ ’).siblings(' ')；

当前元素的父级元素

.parent( )

当前元素内的所有li元素（包括孙子）

.find（‘li’）

当前元素的前一个兄弟元素

.prev( )

当前元素的前面的所有兄弟元素

.prevAll( ‘选择器’)

当前元素的后一个兄弟元素

.next( )

当前元素的后面的所有兄弟元素

.nextAll( )

| **名称** | **用法** | **描述** |
| --- | --- | --- |
| children(selector) | 相当于​('ul-li')，子类选择器 |  |
| find(selector) | 相当于​('ul li'),后代选择器 | 包括孙子 |
| siblings(selector) | $('#first').siblings('li'); | 查找兄弟节点，不包括自己本身。 |
| parent() | $('#first').parent(); | 查找父亲 |
| eq(index) | 相当于​('li:eq(2)'),index从0开始 |  |
| next() | $('li').next() | 找下一个兄弟 |
| prev() | $('li').prev() | 找上一次兄弟 |

### 过滤选择器

获取索引值

.index();

：eq（）

索引值为..的元素 索引值从0开始算

：lt( )

索引值小于..的元素

：gt(　）

索引值大于..的元素

:selected

被选中的 用于表单标签

## 常用方法：

获取body:

document.body

$(document.body) 或者 $（‘body’）

绑定点击事件

$( ).click ( function(){})；

修改内容 任何成对的标签中间的文本内容都可以使用此方法

$( ).text( );

修改样式

$( ).css(‘样式名称’:’值’ );

获取value属性的值 括号内写内容就是设置属性 不用写等号

$(this).val()

修改标签及内容

$( ).html( ‘ <p>aaa</p>‘);

若原来就有标签 想要添加内容则括号里直接写文字

获取焦点的事件

.focus(）

## 补充：

❀

<a href="javascript:void(0);">

不链接，只提供小手 或者在script中的a的点击事件中写return false;

## 链

（$aa）.XX(xxx ) 这种形式下，括号里有内容，返回的还是这个对象，可以用来写链式结构

（$aa）.XX( ) 这种形式下，括号里没有内容 ，返回的是获取的内容

❀

恢复断链：

.end( )

链式编程的原理：

Return this

## 样式的操作

### css操作

❀ 操作单个样式

$('#one').css('background','gray');

❀ 设置多个样式

$('#one').css({

'background':'gray',

'width':'400px',

'height':'200px'

});

❀ 获取样式

$('div').css('background-color');

### class操作

addClass(); 添加 括号里写类名

removeClass(); 移除 括号里要写类名

hasClass(); 是否应用 这个元素有没有添加这个类

toggleClass( ); 切换类 有或无括号内的类名

用法

$obj.addClass(name); name：需要添加的样式类名，注意参数不要带点.

## 基本动画

.slideUp （x,y）

滑入（消失）

.slideDown （x,y）

滑出

.slideToggle(x,y)

滑入滑出切换

.fadeIn

淡入（消失）dfg

.fadeOut

淡出

.fadeToggle

淡入淡出切换

### 自定义动画

$(selector).animate({params},[speed],[easing],[callback]);

{params}：要执行动画的CSS属性，带数字（必选）

speed：执行动画时长（可选）

easing:执行效果，默认为swing（缓动） 可以是linear（匀速）

callback：动画执行完后立即执行的回调函数（可选）

### 其他

显示

.show( X，Y) X：显示所用时间 Y：显示完后调用的方法

隐藏：

.hide() 参数如果是数字 指的是完成效果所用时间（毫秒）

停止当前的动画：

.stop()

stop(clearQueue, jumpToEnd);

第一个参数：是否清除队列

第二个参数：是否跳转到最终效果

## 节点操作

### 添加节点

❀ 向父级元素添加子集元素

$(父级元素) .append(子集元素)

$（子级元素）.appendTo($(父级元素))

子集元素可以写成"<a></a>" 形式

$("<a href='www.baidu.com'>百度</a>").appendTo($("#dv"));

注： appendTo 和 append中 若被添加的元素是已经存在的元素 则为剪切（原来的被删除）

若想复制 可获取子元素的文本内容将内容添加到父级元素 而不是将子元素添加到父级元素

❀ 添加兄弟元素：

添加到后面

$(旧元素).after($(新元素))

添加到前面

$(旧元素).before($(新元素)) 别忘了括号

### 删除节点

$('div').empty(); 清空指定节点的所有元素，自身保留（推荐使用，会清除子元素上绑定的内容，源码）

$('div').html(''); dv下的内容全部删除 包括标签 不推荐使用，会造成内存泄漏，绑定的事件不会被清除。

$("#dv>p").remove(); 自杀 删除自己和后代

### 克隆节点

$(‘子元素’).clone( ).appendTo($(‘’父元素))

注：Id会相同

## 属性操作

### attr

获取属性值

.attr(‘属性名字’) 获取的是属性的值

修改属性值

.attr(‘属性名字，属性值’) 设置单个属性值 可以自定义属性

$('img').attr({ 设置一组属性值

title : '哎哟，不错哦' ,

alt : '哎哟，不错哦' ,

style : 'opacity:.5'

});

移除属性

$('img').removeAttr('title');

### val

获取value值

.val( ) ----表单标签

修改value值

.val ( value值)

### prop 操作布尔属性

用来获取表单属性 比attr更准确

获取属性

.prop( ) 返回的是布尔值

修改属性

.prop(属性，值) 值为布尔类型

$(':checked').prop('checked',true);

### data 操作自定义属性

.data( ) 不传值 获取全部

.data(‘userName’) 传一个值 传什么值获取什么参数

.data(‘userName’,’jinx’) 传两个值是修改

jquery操作的是内存 在html结构上看不到

区别：

设置元素的id属性 用attr 用来操作自定义属性

设置表单属性 选没选中 用prop 用来操作自带属性

## 尺寸和位置操作

### 获取元素宽高

只获取内容宽高：

// 带参数表示设置高度

$('img').height(200);

// 不带参数获取高度

$('img').height();

// 获取可视区宽度

$(window).width();

// 获取可视区高度

$(window).height();

获取元素的宽、高

innerWidth()/innerHeight() 内容+内边距

outerWidth()/outerHeight() 内容+内边距+边框

outerWidth(true)/outerHeight(true) 内边距+边框+外边距

### scrollTop scrollLeft

获取页面被卷曲的高度

$(window).scrollTop();

获取页面被卷曲的宽度

$(window).scrollLeft();

### offset position

获取元素距离document的位置,返回值为对象：{left:100, top:100}

$(selector).offset();

获取相对于其最近的有定位的父元素的位置。

$(selector).position();

## 绑定事件

bind和delegate 都是通过on实现的

delegate和on二者参数顺序不同

### bind 了解

语法:  
对象.bind("事件类型",事件处理函数); （前面放需要绑定事件的对象）  
对象.bind({ "事件类型" : 事件处理函数 ,

"事件类型" : 事件处理函数

});

缺点：不支持动态事件绑定

### delegate 了解

第一个参数：要绑定事件的元素（选择器）

第二个参数：事件类型

第三个参数：事件处理函数

$('.parentBox').delegate('p', 'click', function(){

为 .parentBox下面的所有的p标签绑定事件

});

特点：在父元素中添加新的元素，新的子元素也是可以绑定事件的

缺点：只能注册委托事件（只能给下一代注册事件 第一个参数不能省略）

### on 重点

既能给自己注册事件 又能给子代注册委托事件

on注册简单事件 不支持动态绑定

$(selector).on( 'click', function() {});

on注册事件委托 支持动态绑定

$(selector).on( 'click','span', function() {});

## 解除绑定

用什么方式绑定事件 就用什么方式解绑事件

### unbind

$( selector ).unbind(‘ ‘) 没写参数是指解绑所有事件

$( selector ).unbind(‘ xx xx‘) 解绑指定事件 不同事件用空格隔开

### Undelegate

只能操作子元素

$( selector ).undelegate(); // 解绑所有的子元素事件

$( selector).undelegate( 'click' ); // 解绑指定事件 不同事件用空格隔开

### off 重点

$(selector).off(); 解绑父元素和子元素的所有事件

$(selector).off(‘事件名称’,’选择器’); 不同事件用空格隔开

例子：

$(selector).off('click',’p’); 解绑p的所有click事件 事件名称不能省略

$(selector).off('click'); 解绑父元素和子元素的所有click事件

$(selector).off('click',’\*\*’); 解绑所有的子元素的click事件

### 阻止事件冒泡

return false;

## 触发事件

$(selector).click(); 触发 click事件

$(selector).trigger('click');

作用：可以调用其他元素的事件 但不会触发浏览器的默认行为

## 多库共存

jQuery使用作为标示符，但是如果与其他框架中的​冲突时，jQuery可以释放$符的控制权.

var c = $.noConflict(); // 释放$的控制权,并且把$的能力给了c

## UI

引入ui.css ui.js

引入html css代码的方法：

1. 打开原网页检查 找到关键词

2. ctrl+f 查找所有有关代码 复制 然后再进一步筛选