jQuery由美国人John Resig创建，至今已吸引了来自世界各地的众多JavaScript高手加入其team。

jQuery是继prototype之后又一个优秀的JavaScript框架。其宗旨是——WRITE LESS,DO MORE,写更少的代码,做更多的事情。

它是轻量级的js库，兼容CSS3，还兼容各种浏览器。jQuery是一个快速的，简洁的javaScript库，使用户能更方便地处理HTML/CSS DOM、事件、实现动画效果，并且方便地为网站提供AJAX交互。

jQuery还有一个比较大的优势是，它的文档说明很全，而且各种应用也说得很详细，同时还有许多成熟的插件可供选择。

# 一 jQuery基本用法

jQuery的版本分为1.x系列和2.x、3.x系列，1.x系列兼容低版本的浏览器，2.x、3.x系列放弃支持低版本浏览器，相对来说，目前使用较多的是1.x系列的。

## 1 引入jQuery库

<script type="text/javascript" src="/js/jquery-3.1.1.min.js" ></script>

或者

<script type="text/javascript" src="/js/jquery-3.1.1.js" ></script>

jquery是原始文件，较大，但方便阅读

jquery.mini进行过特殊的处理,如变量的名称基本都写成一个字母,而且格式缩进都被删除了，文件小，但无法方便阅读

## 2 jQuery基本选择器

$(“#id”) :根据id属性

等价于 document.getElementById("id")

$(“tagName”):根据标签名

等价于document.getElementsByTagName("tagName")

$(“.class”) :根据class属性

等价于 document.getElementsByClassName("className")

## 3 jQuery对象与JavaScript对象

jQuery 对象就是通过jQuery包装DOM对象后产生的对象。

jQuery 对象是 jQuery 独有的. 如果一个对象是 jQuery 对象, 那么它就可以使用 jQuery 里的方法: 比如$(“#test”).html();表示获取ID为test的元素内的html代码。

虽然jQuery对象是包装DOM对象后产生的，但是jQuery对象无法使用DOM对象的任何方法，同理DOM对象也不能使用jQuery里的方法。

1）js对象转成jQuery对象

语法：$(js对象)---->jQuery对象

2）jQuery对象转成js对象

语法1：jQuery对象[下标，从0开始]

语法2：jQuery对象.get(下标，从0开始)

# 二 jQuery选择器

## 层次选择器

//获得所有form标签下的所有input标签元素

alert($("form input").length);//4

alert($("#f1 input").length);//3

//获得f3的直接的input子元素

alert($("#f3>input").length);//2

## 增强选择器

//获得第一个input子元素

//val() 获取选择的元素的值，相当于js中的value

alert($("#f1 input:first").val())

//获取最后一个子元素

alert($("#f1 input:last").val());

//根据指定索引获取子元素，子元素索引从0开始

alert($("#f1 input:eq(1)").val());

//gt 获取大于指定索引的元素，

//如果获得了多个元素，val()表示获取查找到的第一个元素得值

alert($("#f1 input:gt(0)").val());

//lt 获取小于指定索引的元素

alert($("#f1 input:lt(1)").val());

//odd索引为奇数的元素，索引从0开始

//css(),设置元素的css样式

$("table tr:odd").css("background-color", "aquamarine");

//even索引为偶数的元素

$("table tr:even").css("background-color", "bisque");

$("table tr:first").css("background-color", "blue")

.css("color", "yellow");

//不会取消原来的样式

//$("table tr:first").css("color", "yellow");

//获得所有的标题标签 （h1-h6）

$(":header").css("color", "red");

## 属性选择器

//html()相当于innerHtml

alert($("div[id='d1']").html());

alert($("input[name='age']").val());

//获取有id属性的div元素

alert($("div[id]").length);

//设置html内容

$("div[id='d1']").html("<b>hahaha</b>");

$("#d1").text("hahaha");

$("input[name='age']").val(30);

## 表单选择器

// :input 包括input、select、textarea

alert($(":input").length);//7

//input标签

alert($("input").length);//5

//针对type

alert($(":text").length);//1

alert($(":password").length);

alert($(":radio").length);

alert($(":checkbox").length);

alert($(":submit").length);

## 表单对象属性选择器

//选中状态

//alert($(":checkbox:checked").length);

//alert($(":checkbox:checked").val());

//非选中状态

//alert($(":checkbox:NOT(:checked)").length);

//获得选中的option

alert($("option:selected").text());

alert($("option:selected").val());

//非选中状态 如果使用html(),只会显示第一个元素的内容

//如果使用text() 会显示所有的元素的内容

alert($("option:not(:selected)").html());

alert($("option:not(:selected)").text());

# 三 jQuery常用方法

each()：是jQuery中专用于迭代数组、对象的方法，参数为一个处理函数，this表示当前需要迭代的js对象

//数组属于js对象

var arr = [1, 2, 3, 4, 5];

//jQery对象.each()遍历

$(arr).each(function(){

//this 表示遍历的每个元素，this是js对象

//alert(this);

});

//each中的函数的参数：index 表示索引，item表示遍历的元素

$(arr).each(function(index, item){

//alert(index);

//alert(item);

});

//遍历

//each的第一个参数可以是js对象或jQuery对象

$.each(arr, function(){

//alert(this);

});

$.each(arr, function(index, item){

//alert(index);

});

$.each($(arr), function(){

//alert(this);

});

var obj = [{name:"zhagnsan", age:10}, {name:"lisi", age:20}];

$(obj).each(function(){

//alert(this.name);

//alert(this.age);

});

var obj1 = {name:"zhangsan", age:10};

//遍历对象时，function的参数表示对象中元素的key值和value值

$.each(obj1, function(key, value) {

//alert(key);

//alert(value);

});

</script>

$("<div id='d1'>haha</div>")：创建元素，属性，文本

remove()：删除自已及其后代节点

append()：追加到父元素之后

prepend()：追加到父元素之前

after()：追加到兄弟元素之后

before()：追加到兄弟元素之前

<script>

//创建DOM对象

var $div = $("<div>hello</div>");

var $p = $("<p><font color='red'>world</font></p>")

//document.createElement("div");

//在最后添加子元素

$("body").append($div);

//在最前面添加子元素

$("body").prepend($p);

//删除元素

//$("table").remove();

//删除元素下的所有子元素

//$("body").empty();

</script>

attr(name)：获取属性值

attr(name,value)：给符合条件的标签添加属性

removeAttr()：删除已存在的属性

<script>

//获取元素属性对象的值

//alert($("table").attr("width"));

//设置属性的值

$("table").attr("width", "300");

//$("table").removeAttr("border");

//prop()获取、设置属性的值

//alert($("table").prop("border"));

</script>

val()：获取value属性的值

val("")：设置value属性值为""空串，相当于清空

text()：获取HTML或XML标签之间的值

text("")：设置HTML或XML标签之间的值为""空串

addClass()：增加已存在的样式

removeClass()：删除已存在的样式

hasClass()：判断标签是否有指定的样式，true表示有样式，false表示无样式

toggleClass()：如果标签有样式就删除，否则增加样式

<script>

//添加class属性值

$("#d6").addClass("divC");

$("#d5").removeClass("divC");

function btnClick(){

//如果原来有class属性，删除，如果没有，加上

$("div").toggleClass("divC");

}

</script>

offset()：获取对象的left和top坐标

offset({top:100,left:200})：将对象直接定位到指定的left和top坐标

width()：获取对象的宽

width(300)：设置对象的宽

height()：获取对象的高

height(500)：设置对象的高

<script>

//获得元素的宽度

//alert($("table").width());

//设置宽度

$("table").width(200);

//设置高度

$("table").height(80);

//获取浏览器空白区域的宽度、高度

alert($(window).width());

</script>

show()：显示对象

hide()：隐藏对象

fadeIn()：淡入显示对象

fadeOut()：淡出隐藏对象

slideUp()：向上滑动

slideDown()：向下滑动

slideToggle()：上下切换滑动

# 四 jQuery常用事件

ready：在浏览器加载web页面时触发，可以写多次ready事件，不会后者覆盖前者，依次从上向下执行，我们常用$(函数)简化

<script>

$(document).ready(function(){

alert("页面加载完成");

});

//简化的写法

$(function(){

alert("页面加载完成2");

});

</script>

click：单击触发

$("#btn1").click(function(){

alert("点我干啥");

});

dblclick：双击触发

var flag = 0;

//双击事件

$("div").dblclick(function(){

if(flag == 0){

$(this).css("background-color", "yellow");

flag = 1;

}else{

$(this).css("background-color", "white");

flag = 0;

}

});

change：当内容改变时触发

focus：焦点获取

blur：焦点失去

select：选中所有的文本值

$(function(){

//让元素得到焦点

$("#in1").focus();

//获得焦点时触发的事件

$("#in2").focus(function(){

//$(this).select();

$(this).val("");

});

});

keyup/keydown/keypress：

mousemove：在指定区域中不断移动触发

mouseover：鼠标移入时触发

mouseout：鼠标移出时触发

//鼠标移动到元素上是触发事件

$("table tr").mouseover(function(){

$(this).css("background-color", "yellow");

});

//鼠标离开元素触发事件

$("table tr").mouseout(function(){

$(this).css("background-color", "white");

});

hover：该方法有两个参数，第一个表示鼠标移动到元素上时触发事件的函数，第二个表示离开时触发事件的函数

$("table tr").hover(function(){

$(this).css("background-color", "yellow");

}, function(){

$(this).css("background-color", "white");

})

submit：在提交表单时触发，true表示提交数据，false表示不提交

<script>

//表单的提交事件

//如果返回false，不会向后台提交数据

$("form").submit(function(){

var v = $(":text").val();

//去除字符串两端的空白字符

v = $.trim(v);

if(v.length == 0){

$("#s1").text("必须输入姓名");

return false;

}

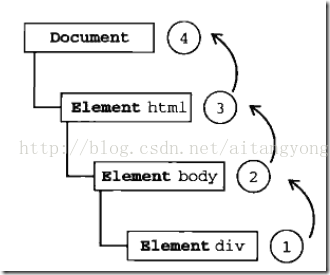
});

</script>

# 五 事件冒泡

在一个对象上触发某类事件（比如单击onclick事件），如果此对象定义了此事件的处理程序，那么此事件就会调用这个处理程序，如果没有定义此事件处理程序或者事件返回true，那么这个事件会向这个对象的父级对象传播，从里到外，直至它被处理（父级对象所有同类事件都将被激活），或者它到达了对象层次的最顶层，即document对象、window对象。

事件冒泡类似：你在地方法院要上诉一件案子，如果地方没有处理此类案件的法院，地方相关部门会帮你继续往上级法院上诉，比如从市级到省级，直至到中央法院，最终使你的案件得以处理。



阻止冒泡

event.stopPropagation();

阻止默认行为

event.preventDefault();

<script type="text/javascript">

$(function(){

$('body').click(function() {

alert(4);

});

$('.grandfather').click(function() {

alert(3);

});

$('.father').click(function() {

alert(2);

});

$('.son').click(function(ev) {

alert(1);

//阻止冒泡

//ev.stopPropagation();

//ev.preventDefault();

//console.log(ev);

alert(ev.clientX);

//return false;

});

$(document).click(function(event){

alert("doc")

})

$(window).click(function(event){

alert("win")

})

//不显示点击鼠标右键时弹出的菜单

$(document).contextmenu(function(event) {

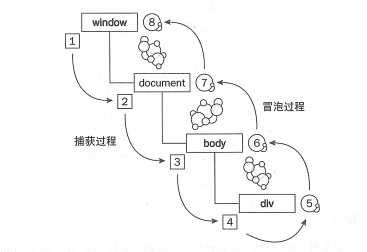
//event.preventDefault();

return false;

});

});

关于事件捕获、冒泡



事件的传递过程包含三个阶段，分别称为：捕获阶段，目标阶段，冒泡阶段

1.捕获阶段：先由文档的根节点document往事件触发对象，从外向内捕获事件对象；

2.目标阶段：到达目标事件位置（事发地），触发事件；

3.冒泡阶段：再从目标事件位置往文档的根节点方向回溯，从内向外冒泡事件对象。

DOM方法 addEventListener() 和 removeEventListener()是用来分配和删除事件的函数。

这两个方法都需要三个参数，分别为：

事件名称（String）

要触发的事件处理函数(Function)

useCapture（**指定事件处理函数的时期或阶段(boolean)）**。

<script type="text/javascript">

function xx1(){ //蓝

alert(123);

}

function xx2(){ //红

alert(345);

}

function xx3(){//黄

alert(456);

}

window.onload = function(){

var oDvi1 = document.getElementById('div1');

var oDvi2 = document.getElementById('div2');

var oDvi3 = document.getElementById('div3');

// 如果为true,表示事件捕获 123 -> 345 -> 456

// 如果为fase，表示事件冒泡 456-> 345 ->123

oDvi1.addEventListener('click', xx1, false);

oDvi2.addEventListener('click', xx2, false);

oDvi3.addEventListener('click', xx3, false);

}

</script>

在div3上触发点击事件后：

**捕获阶段**: 外->里 \* 在div1处检测 useCapture 是否为true,是则执行程序,div2同理 .

**目标阶段**: 目标到div3处，发现div3就是鼠标点击的节点， 所以这里是目标阶段。若有事件处理程序，则执行该程序，这里不论 useCapture 为 true 还是 false。

**冒泡阶段**: 里->外  在div2处检测useCapture 是否为false， 是则执行该程序 . div1同理

实例：自定义弹窗

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<style type="text/css">

.pop\_con{

display:none;

}

.pop{

width:400px;

height:300px;

background-color:#fff;

border:1px solid #000;

position:fixed;

left:50%;

top:50%;

margin-left:-200px;

margin-top:-150px;

z-index:9999;

}

.mask{

position:fixed;

width:100%;

height:100%;

background-color:#000;

left:0;

top:0;

opacity:0.3;

filter:alpha(opacity=30);

z-index:9990;

}

</style>

<script type="text/javascript" src="js/jquery-1.12.2.min.js"></script>

<script type="text/javascript">

$(function(){

$('#btn').click(function() {

$('#pop').show();

return false;

});

$('#shutoff').click(function() {

alert(1);

$('#pop').hide();

});

$(document).click(function(event) {

alert(2)

$('#pop').hide();

});

//不显示点击鼠标右键时弹出的菜单

// $(".pop").contextmenu(function(event) {

// //event.preventDefault();

// return false;

// });

// $(".mask").contextmenu(function(event) {

// //event.preventDefault();

// return false;

// });

//为了鼠标右键任何元素都不显示，直接设置document

// contextmenu 小写的，hbuilder联想不出来

// 不能使用event.stopPropagation();

$(document).contextmenu(function(event) {

//event.preventDefault();

return false;

});

})

</script>

</head>

<body>

<h1>首页标题</h1>

<p>页面内容</p>

<input type="button" name="" value="弹出" id="btn">

<div class="pop\_con" id="pop">

<div class="pop">

<h3>提示信息！</h3>

<a href="#" id="shutoff">关闭</a>

<input type="text" name="">

</div>

<div class="mask"></div>

</div>

</body>

</html>