**二十一 、javascript-JQuery（下）**

**一、主动触发**

**1.js中的触发**

oBtn.onclick = function(){

alert(123);

}

**onclick事件函数是由系统帮助我们调用的，成为被动触发**

【注】主动触发 函数名();

oBtn.onclick();

**2.JQuery中的主动触发**

trigger 用于主动触发某一个节点上的某一个事件

【注】trigger可以触发系统提供事件类型函数

【注】trigger可以触发用户自定义的事件。

$("#btn").on("show", function(){

alert(456);

})

$("#btn").trigger('show');

【注】对于用户自定义事件等场景是可以用到的。

**二、事件委托**

delegate 代理

undelegate

delegate('selector', 'eventType', function(event) {

selector

});

第一个参数： 代理对象 支持CSS选择器选择元素

第二个参数： 事件类型

第三个参数： 绑定的函数

$("#ul1").delegate("li,div", "click", function(){

$(this).css("backgroundColor", "red");

$("#ul1").undelegate('li,div', 'click');

})

$("#ul1").delegate("div", "click", function(){

$(this).css("backgroundColor", "red");

})

【注】应用undelegate 时，第一个参数所传的代理对象需要和delegate的一致

**三、工具方法**

$().xxx() $().yyy() JQ对象的方法，只能被JQ对象调用

$.xxx() $.yyy() 工具方法/和普通函数一样 也可以被JS所使用

**1.$.type() 相当于typeof() 判断当前数据类型**

var a = /box/;

alert(typeof a); //object

alert($.type(a)); //regexp

【**$.type()** 】要更加精确

**2.$.trim() 删除首尾空格**

var str = " he ll o ";

alert("|" + str.trim() + "|");//|he ll o|

alert("|" + $.trim(str) + "|");//|he ll o|

**3.inArray() 相当于 indexOf()**

第一个参数：查找的元素

第二个参数：数组

var arr = ["a", "b", "c", "d"];

alert($.inArray("b", arr));

**4.$.proxy() 改变this的指向**

function show(n1, n2){

alert(this);

alert(n1 + ", " + n2);

}

//直接调用，第一个参数为函数，第二个参数为this指向

$.proxy(show, document)(3, 4);

$.proxy(show, document, 3, 4)();

//赋值

var func = $.proxy(show, document, 10, 20);

func();

**四、noConflict**

var Tian = $.noConflict(); //给JQ起一个别名，避免冲突，以防var $ = 10;这样的事情发生

var $ = 10;

alert(typeof Tian); //function

Tian(function(){

alert(123);

})

**五、parseJSON**

**1.parseJSON 将字符串转化成对应数据结构**

var str = '{"name":"hello"}';

alert($.parseJSON(str).name);

**2.makeArray 转化成数组**

window.onload = function(){

var aDivs = document.getElementsByTagName("div");

//aDivs并不是数组 伪数组

// aDivs.push();

aDivs = $.makeArray(aDivs);

aDivs.push();

}

**六、插件方法**

**【注】如果JQ再有这么一个方法就更好了。**

**【注】通过给JQ添加插件方法完成。**

**JQ拓展接口**

**（1）$.extend(); 拓展的是工具方法 $.xxx();**

**（2）$.fn.extend();拓展的是JQ对象的方法 $().xxx()**

**通过插件方式进行拓展（两种写法）**

**（1）$.fn.extend(函数名, 函数);**

**（2）$.fn.extend(json对象)**

**【例1】拖拽函数**

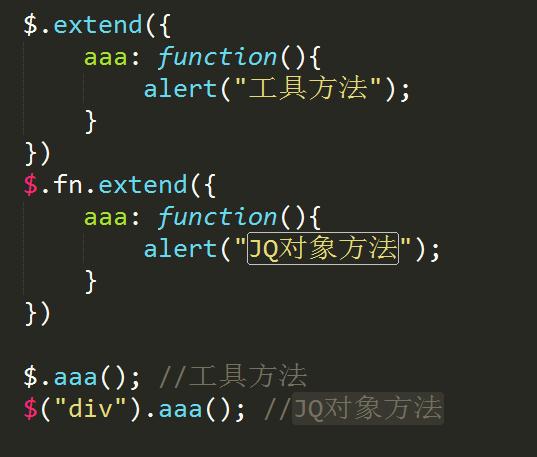


**调用：$("div").drag();**

**【例2】添加一个删除首空格的字符串函数**

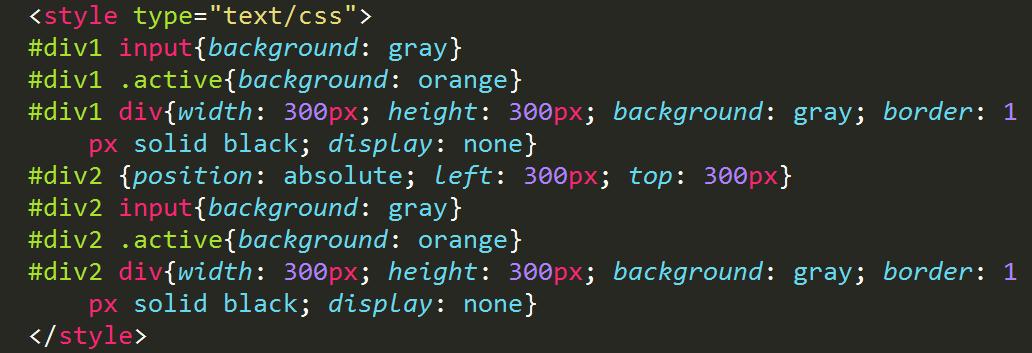


【注】工具方法和JQ对象方法

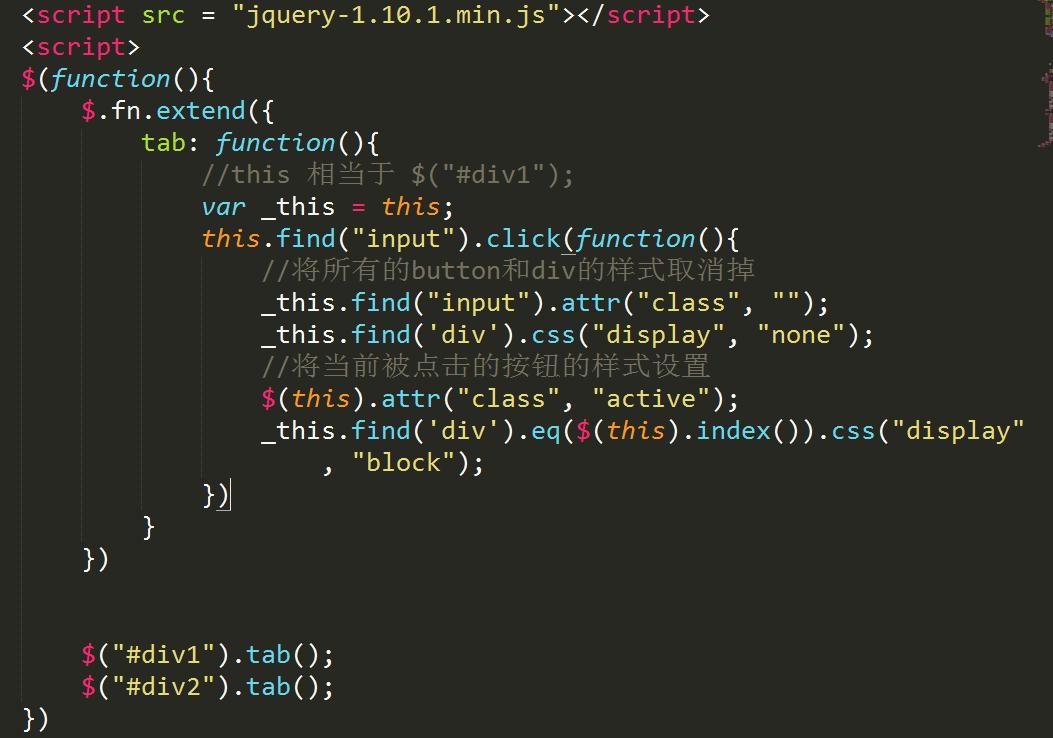


**【综合实例】选项卡**

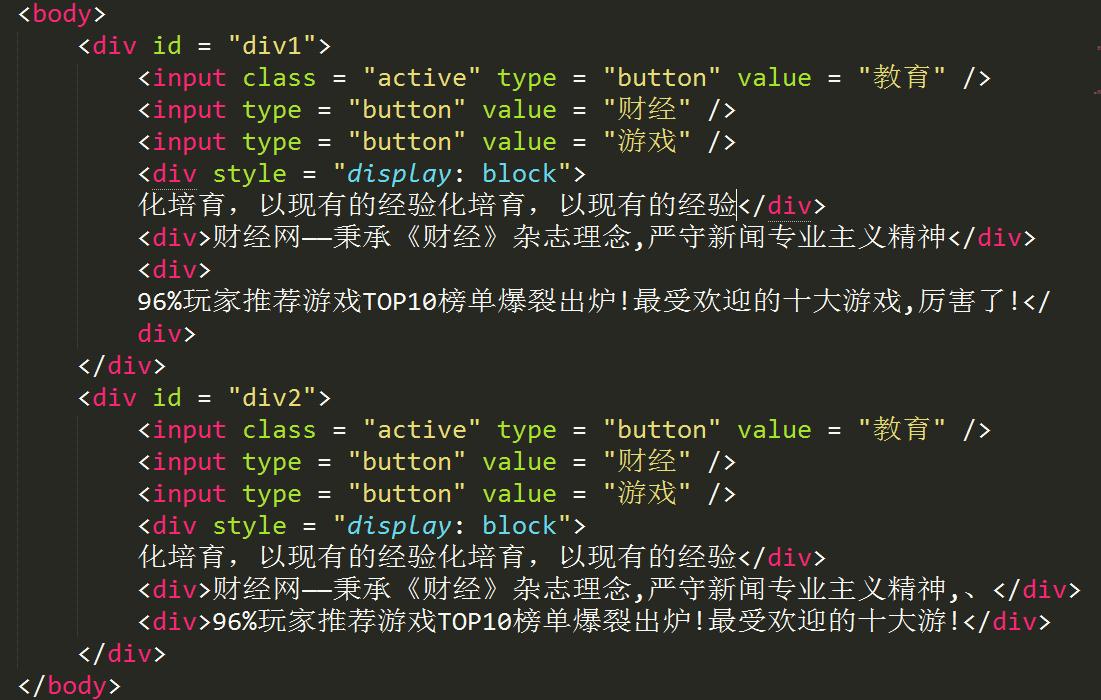
**【style部分】**



**【script部分】**



**【body部分】**



**六、JQuery\_AJAX**

**1.AJAX\_load**

下载数据，并且将数据填充在前面被选中元素的innerHTML

【注】我们可以对下载到的数据进行筛选。

【格式】$(selector).load(URL,data,callback);

==============================================

【1.txt】

<h2>jQuery and AJAX is FUN!!!</h2>

<p id="p1">This is some text in a paragraph.</p>

===============================================

//下载1.txt中的数据（h2与p都被解析并添加到div1中）

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

$("#div1").load("1.txt", function(data){

alert("下载完成了:" + data);

})

})

//下载1.txt中h2的数据（只有h2中的数据被解析并添加到div1中）

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

$("#div1").load("1.txt h2", function(data){

alert("下载完成了:" + data);

})

})

**2.AJAX\_函数**

（1）$.get

$.get('1.get.php', {

username: "xxx",

age: 18,

password: 123456

}, function(data) {

alert(data);

});

（2）$.post

$.post('1.post.php', {

username: "xxx",

age: 18,

password: 123456

}, function(data) {

alert(data);

});

（3）$.ajax

$.ajax({

url: '/path/to/file',

type: 'default GET (Other values: POST)',

dataType: 'default: Intelligent Guess (Other values: xml, json, script, or html)', //自动解析数据类型

data: {param1: 'value1'},

})

.done(function() {

console.log("success");

})

.fail(function() {

console.log("error");

})

.always(function() {

console.log("complete");

});

**七、JQuery\_cookie**

//设置cookie

//raw: true 代表将编码解码关闭，为false时默认编码解码开启，将汉字转化为字符，防止乱码

$.cookie("变种人", "X教授", {expires: 7, raw: false});

$.cookie("超级英雄", "钢铁侠", {expires: 7, raw: false});

$.cookie("变形金刚", "擎天柱", {expires: 7, raw: true});

alert($.cookie("变种人")); //X%E6%95%99%E6%8E%88

alert($.cookie("超级英雄")); // %E9%92%A2%E9%93%81%E4%BE%A0

alert($.cookie("变形金刚")); // 擎天柱

$.cookie("超级英雄", null);

alert($.cookie("超级英雄"));//null