var reClass = new RegExp("(^| )" + sClass + "( |$)");

(^| ) 开始什么都没有，或者，一个空格

( |$) 结束什么都没有，或者，一个空格

^ 以。。。开始

| 或者

$ 以。。。结束

var get = {

byId: function(id) {

return typeof id === "string" ? document.getElementById(id) : id

},

byClass: function(sClass, oParent) {

var aClass = [];

var reClass = new RegExp("(^| )" + sClass + "( |$)");

var aElem = this.byTagName("\*", oParent);

for (var i = 0; i < aElem.length; i++) reClass.test(aElem[i].className) && aClass.push(aElem[i]);

return aClass

},

byTagName: function(elem, obj) {

return (obj || document).getElementsByTagName(elem)

}

};

/\*-------------------------- +

事件绑定, 删除

+-------------------------- \*/

var EventUtil = {

addHandler: function (oElement, sEvent, fnHandler) {

oElement.addEventListener ? oElement.addEventListener(sEvent, fnHandler, false) : (oElement["\_" + sEvent + fnHandler] = fnHandler, oElement[sEvent + fnHandler] = function () {oElement["\_" + sEvent + fnHandler]()}, oElement.attachEvent("on" + sEvent, oElement[sEvent + fnHandler]))

},

removeHandler: function (oElement, sEvent, fnHandler) {

oElement.removeEventListener ? oElement.removeEventListener(sEvent, fnHandler, false) : oElement.detachEvent("on" + sEvent, oElement[sEvent + fnHandler])

},

addLoadHandler: function (fnHandler) {

this.addHandler(window, "load", fnHandler)

}

};

/\*-------------------------- +

设置css样式

读取css样式

+-------------------------- \*/

function css(obj, attr, value)

{

switch (arguments.length)

{

case 2:

if(typeof arguments[1] == "object")

{

for (var i in attr) i == "opacity" ? (obj.style["filter"] = "alpha(opacity=" + attr[i] + ")", obj.style[i] = attr[i] / 100) : obj.style[i] = attr[i];

}

else

{

return obj.currentStyle ? obj.currentStyle[attr] : getComputedStyle(obj, null)[attr]

}

break;

case 3:

attr == "opacity" ? (obj.style["filter"] = "alpha(opacity=" + value + ")", obj.style[attr] = value / 100) : obj.style[attr] = value;

break;

}

};

EventUtil.addLoadHandler(function ()

{

var oMsgBox = get.byId("msgBox");

var oUserName = get.byId("userName");

var oConBox = get.byId("conBox");

var oSendBtn = get.byId("sendBtn");

var oMaxNum = get.byClass("maxNum")[0];

var oCountTxt = get.byClass("countTxt")[0];

var oList = get.byClass("list")[0];

var oUl = get.byTagName("ul", oList)[0];

var aLi = get.byTagName("li", oList);

var aFtxt = get.byClass("f-text", oMsgBox);

var aImg = get.byTagName("img", get.byId("face"));

var bSend = false;

var timer = null;

var oTmp = "";

var i = 0;

var maxNum = 140;

//禁止表单提交

EventUtil.addHandler(get.byTagName("form", oMsgBox)[0], "submit", function () {return false});

//为广播按钮绑定发送事件

EventUtil.addHandler(oSendBtn, "click", fnSend);

//为Ctrl+Enter快捷键绑定发送事件

EventUtil.addHandler(document, "keyup", function(event)

{

var event = event || window.event;

event.ctrlKey && event.keyCode == 13 && fnSend()

});

//发送广播函数

function fnSend ()

{

var reg = /^\s\*$/g;

if(reg.test(oUserName.value))

{

alert("\u8bf7\u586b\u5199\u60a8\u7684\u59d3\u540d");

oUserName.focus()

}

else if(!/^[u4e00-\u9fa5\w]{2,8}$/g.test(oUserName.value))

{

alert("\u59d3\u540d\u75312-8\u4f4d\u5b57\u6bcd\u3001\u6570\u5b57\u3001\u4e0b\u5212\u7ebf\u3001\u6c49\u5b57\u7ec4\u6210\uff01");

oUserName.focus()

}

else if(reg.test(oConBox.value))

{

alert("\u968f\u4fbf\u8bf4\u70b9\u4ec0\u4e48\u5427\uff01");

oConBox.focus()

}

else if(!bSend)

{

alert("\u4f60\u8f93\u5165\u7684\u5185\u5bb9\u5df2\u8d85\u51fa\u9650\u5236\uff0c\u8bf7\u68c0\u67e5\uff01");

oConBox.focus()

}

else

{

var oLi = document.createElement("li");

var oDate = new Date();

oLi.innerHTML = "<div class=\"userPic\"><img src=\"" + get.byClass("current", get.byId("face"))[0].src + "\"></div>\

<div class=\"content\">\

<div class=\"userName\"><a href=\"javascript:;\">" + oUserName.value + "</a>:</div>\

<div class=\"msgInfo\">" + oConBox.value.replace(/<[^>]\*>|&nbsp;/ig, "") + "</div>\

<div class=\"times\"><span>" + format(oDate.getMonth() + 1) + "\u6708" + format(oDate.getDate()) + "\u65e5 " + format(oDate.getHours()) + ":" + format(oDate.getMinutes()) + "</span><a class=\"del\" href=\"javascript:;\">\u5220\u9664</a></div>\

</div>";

//插入元素

aLi.length ? oUl.insertBefore(oLi, aLi[0]) : oUl.appendChild(oLi);

//重置表单

get.byTagName("form", oMsgBox)[0].reset();

for (i = 0; i < aImg.length; i++) aImg[i].className = "";

aImg[0].className = "current";

//将元素高度保存

var iHeight = oLi.clientHeight - parseFloat(css(oLi, "paddingTop")) - parseFloat(css(oLi, "paddingBottom"));

var alpah = count = 0;

css(oLi, {"opacity" : "0", "height" : "0"});

timer = setInterval(function ()

{

css(oLi, {"display" : "block", "opacity" : "0", "height" : (count += 8) + "px"});

if (count > iHeight)

{

clearInterval(timer);

css(oLi, "height", iHeight + "px");

timer = setInterval(function ()

{

css(oLi, "opacity", (alpah += 10));

alpah > 100 && (clearInterval(timer), css(oLi, "opacity", 100))

},30)

}

},30);

//调用鼠标划过/离开样式

liHover();

//调用删除函数

delLi()

}

};

//事件绑定, 判断字符输入

EventUtil.addHandler(oConBox, "keyup", confine);

EventUtil.addHandler(oConBox, "focus", confine);

EventUtil.addHandler(oConBox, "change", confine);

//输入字符限制

function confine ()

{

var iLen = 0;

for (i = 0; i < oConBox.value.length; i++) iLen += /[^\x00-\xff]/g.test(oConBox.value.charAt(i)) ? 1 : 0.5;

oMaxNum.innerHTML = Math.abs(maxNum - Math.floor(iLen));

maxNum - Math.floor(iLen) >= 0 ? (css(oMaxNum, "color", ""), oCountTxt.innerHTML = "\u8fd8\u80fd\u8f93\u5165", bSend = true) : (css(oMaxNum, "color", "#f60"), oCountTxt.innerHTML = "\u5df2\u8d85\u51fa", bSend = false)

}

//加载即调用

confine();

//广播按钮鼠标划过样式

EventUtil.addHandler(oSendBtn, "mouseover", function () {this.className = "hover"});

//广播按钮鼠标离开样式

EventUtil.addHandler(oSendBtn, "mouseout", function () {this.className = ""});

//li鼠标划过/离开处理函数

function liHover()

{

for (i = 0; i < aLi.length; i++)

{

//li鼠标划过样式

EventUtil.addHandler(aLi[i], "mouseover", function (event)

{

this.className = "hover";

oTmp = get.byClass("times", this)[0];

var aA = get.byTagName("a", oTmp);

if (!aA.length)

{

var oA = document.createElement("a");

oA.innerHTML = "删除";

oA.className = "del";

oA.href = "javascript:;";

oTmp.appendChild(oA)

}

else

{

aA[0].style.display = "block";

}

});

//li鼠标离开样式

EventUtil.addHandler(aLi[i], "mouseout", function ()

{

this.className = "";

var oA = get.byTagName("a", get.byClass("times", this)[0])[0];

oA.style.display = "none"

})

}

}

liHover();

//删除功能

function delLi()

{

var aA = get.byClass("del", oUl);

for (i = 0; i < aA.length; i++)

{

aA[i].onclick = function ()

{

var oParent = this.parentNode.parentNode.parentNode;

var alpha = 100;

var iHeight = oParent.offsetHeight;

timer = setInterval(function ()

{

css(oParent, "opacity", (alpha -= 10));

if (alpha < 0)

{

clearInterval(timer);

timer = setInterval(function ()

{

iHeight -= 10;

iHeight < 0 && (iHeight = 0);

css(oParent, "height", iHeight + "px");

iHeight == 0 && (clearInterval(timer), oUl.removeChild(oParent))

},30)

}

},30);

this.onclick = null

}

}

}

delLi();

//输入框获取焦点时样式

for (i = 0; i < aFtxt.length; i++)

{

EventUtil.addHandler(aFtxt[i], "focus", function () {this.className = "active"});

EventUtil.addHandler(aFtxt[i], "blur", function () {this.className = ""})

}

//格式化时间, 如果为一位数时补0

function format(str)

{

return str.toString().replace(/^(\d)$/,"0$1")

}

//头像

for (i = 0; i < aImg.length; i++)

{

aImg[i].onmouseover = function ()

{

this.className += " hover"

};

aImg[i].onmouseout = function ()

{

this.className = this.className.replace(/\s?hover/,"")

};

aImg[i].onclick = function ()

{

for (i = 0; i < aImg.length; i++) aImg[i].className = "";

this.className = "current"

}

}

});