**点击按钮加入购物车**

/\*\*

\* addcart 将商品加入购物车

\* @goods\_id 商品id

\* @num 商品数量

\* @form\_id 商品详情页所在的 form表单

\* @to\_catr 加入购物车后再跳转到 购物车页面 默认不跳转 1 为跳转

\*/

function AjaxAddCart(goods\_id,num,to\_catr)

{

// 如果有商品规格 说明是商品详情页提交

if($("#buy\_goods\_form").length > 0){

$.ajax({

type : "POST",

url:"xxx",

data : $('#buy\_goods\_form').serialize(),// 你的formid 搜索表单 序列化提交

dataType:'json',

success: function(data){

if(data.status < 0)

{

layer.alert(data.msg, {icon: 2});

return false;

}

// 加入购物车后再跳转到 购物车页面

if(to\_catr == 1) //直接购买

{

location.href = "/index.php?m=Home&c=Cart&a=cart";

}

else

{

cart\_num = parseInt($('#cart\_quantity').html())+parseInt($('input[name="goods\_num"]').val());

$('#cart\_quantity').html(cart\_num)

layer.open({

type: 2,

title: '温馨提示',

skin: 'layui-layer-rim', //加上边框

area: ['490px', '386px'], //宽高

content:["/index.php?m=Home&c=Goods&a=open\_add\_cart","no"],

success: function(layero, index) {

layer.iframeAuto(index);

}

});

}

}

});

}else{ // 否则可能是商品列表页 收藏页 等点击加入购物车的

$.ajax({

type : "POST",

url:"/index.php?m=Home&c=Cart&a=ajaxAddCart",

data :{goods\_id:goods\_id,goods\_num:num} ,

dataType:'json',

success: function(data){

if(data.status == -1)

{

location.href = "/index.php?m=Home&c=Goods&a=goodsInfo&id="+goods\_id;

}

else

{

// 加入购物车有误

if(data.status < 0)

{

layer.alert(data.msg, {icon: 2});

return false;

}

cart\_num = parseInt($('#cart\_quantity').html())+parseInt(num);

$('#cart\_quantity').html(cart\_num)

layer.open({

type: 2,

title: '温馨提示',

skin: 'layui-layer-rim', //加上边框

area: ['490px', '386px'], //宽高

content:"/index.php?m=Home&c=Goods&a=open\_add\_cart"

});

}

}

});

}

}

/\*前端接受后端json数据\*/

$.ajax({

url: 'xxx?' + Math.random(), // 后面加个随机数是清除缓存

type: 'get',

dataType: 'json',

success: function (user) {

//字符串拼接

var htmlNodes = '';

for(var i = 0; i < user.length; i ++){

htmlNodes +=/\* '<figure><div class="img"><img width="120" height="120" src="' + user[i].images + '"></div><figcaption><strong>' + user[i].name + '</strong><p>' + user[i].caption + '</p></figcaption></figure>'; \*/

}

$('#container').append(htmlNodes);

}

});

**一、原生js通过ajax获取json数据**

原生js创建ajax对象比较复杂，因为IE浏览器对ajax对象的创建于其他浏览器不同，下面创建ajax对象的方法就兼容了不同的浏览器

**[javascript]** [view plain](http://blog.csdn.net/dorry_l/article/details/48738033) [copy](http://blog.csdn.net/dorry_l/article/details/48738033)

* function createXMLHttpRequest(){
* try{
* XMLHttpReq = new ActiveXObject("Msxml2.XMLHTTP");    //在IE创建较高版本的ajax对象
* }
* catch(e){
* try{
* XMLHttpReq = new ActiveXObject("Microsoft.XMLHTTP");//在IE中创建较低版本的ajax对象
* }
* catch (e){
* XMLHttpReq = new XMLHttpRequest(); //其他浏览器ajax对象的创建
* }
* }
* }

创建完ajax对象之后，下面就是发送ajax请求了

**[javascript]** [view plain](http://blog.csdn.net/dorry_l/article/details/48738033) [copy](http://blog.csdn.net/dorry_l/article/details/48738033)

* function sendAjaxRequest() {
* createXMLHttpRequest();
* XMLHttpReq.open('post', 'test.php', true);
* XMLHttpReq.onreadystatechange = response;
* XMLHttpReq.send(null);
* }

**[javascript]** [view plain](http://blog.csdn.net/dorry_l/article/details/48738033) [copy](http://blog.csdn.net/dorry_l/article/details/48738033)

* //接收后台返回的数据

**[javascript]** [view plain](http://blog.csdn.net/dorry_l/article/details/48738033) [copy](http://blog.csdn.net/dorry_l/article/details/48738033)

* function response() {
* if(XMLHttpReq.readyState == 4){
* if(XMLHttpReq.status == 200){
* var user = XMLHttpReq.responseText;
* user = eval(user); //这里需注意，后台返回的是纯文本数据而不是json数据，要用eval方法转换成json格式的数据
* //利用后台返回的json数据自动创建节点，采用拼接字符串的方法
* for(var i = 0; i < user.length; i ++){
* htmlNodes += '<figure><div class="img"><img width="120" height="120" src="' + user[i].images + '"></div><figcaption><strong>' + user[i].name + '</strong><p>' + user[i].caption + '</p></figcaption></figure>';
* }
* $('#container').append(htmlNodes);
* }
* }
* }

**[php]** [view plain](http://blog.csdn.net/dorry_l/article/details/48738033) [copy](http://blog.csdn.net/dorry_l/article/details/48738033)

* <?php
* echo '[
* {   "name" : "Herry",
* "caption" : "还没好好的感受，雪花绽放的气候",
* "images" : "images/1.jpg"
* },
* {
* "name" : "Dorry",
* "caption" : "我们一起颤抖,会更明白什么是温柔",
* "images" : "images/4.jpg"
* },
* {
* "name" : "Money",
* "caption" : "还没跟你牵着手,走过荒芜的沙丘",
* "images" : "images/5.png"
* },
* {
* "name" : "Herry",
* "caption" : "可能从此以后学会珍惜,天长和地久",
* "images" : "images/2.jpg"
* },
* {
* "name" : "Herry",
* "caption" : "有时候有时候,我会相信一切有尽头,相聚离开都有时候,没有什么会永垂不朽",
* "images" : "images/3.jpeg"
* }
* ]';

**二、用jquery通过ajax获取json数据**

**[javascript]** [view plain](http://blog.csdn.net/dorry_l/article/details/48738033) [copy](http://blog.csdn.net/dorry_l/article/details/48738033)

* $.ajax({
* url: 'test.php?' + Math.random(),  // 后面加个随机数是清除缓存
* type: 'get',
* dataType: 'json',
* success: function (user) {
* //字符串拼接
* var htmlNodes = '';
* for(var i = 0; i < user.length; i ++){
* htmlNodes += '<figure><div class="img"><img width="120" height="120" src="' + user[i].images + '"></div><figcaption><strong>' + user[i].name + '</strong><p>' + user[i].caption + '</p></figcaption></figure>';
* }
* $('#container').append(htmlNodes);
* }
* });

效果图



|  |
| --- |
|  |
| **<script>** |
|  | $(document).ready(function() { |
|  | getlist(); |
|  | }); |
|  |  |
|  | function getlist(id) { |
|  | $.ajax({ |
|  | type: "post", |
|  | url: "<{:U('Product/getgoods')}>", |
|  | data: { |
|  | 'id': id |
|  | }, |
|  | async: true, |
|  | dataType: 'json', |
|  | success: function(data) { |
|  | var goods = ''; |
|  | $.each(data, function(k, v) { |
|  | goods += '<li >'; |
|  | goods += '<div class="pro-img"><img class="pro-imgs" src="' + v.product\_img + '"/></div>'; |
|  | goods += '<p><span class="pro-name"><a href="index.php?m=&c=Product&a=productdetail&id=' + v.id + '">' + v.product\_name + '</a></span><br>'; |
|  | goods += '<span class="pro-info"><a href="index.php?m=&c=Product&a=productdetail&id=' + v.id + '">' + v.product\_title + '</a></span><br>'; |
|  | goods += '<a href="index.php?m=&c=Product&a=productdetail&id=' + v.id + '" class="btn more-b">更多<span></span></a></p>'; |
|  | }); |
|  | $('.goods').html(goods); |
|  | } |
|  | }); |
|  | } |
|  | </script> |
|  |  |

**Ajax封装方法**

/\*\*

\* ajax post提交

\* @param url

\* @param param

\* @param datat 为html,json,text

\* @param callback回调函数

\* @return

\*/

function sendAjax(url, param, datat, callback) {

$.ajax({

type: "post",

url: url,

data: param,

dataType: datat,

success: callback,

error: function() {

ClosetoastLoading();

errorTip();

}

});

}

**左侧tab选项卡**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>选项卡</title>

<style type="text/css">

<!--

\* { margin:0; padding:0; font-size:12px; font-weight:normal; }

.jj { font-weight:bolder!important; }

.box { border-top-color:#c00!important; }

.pr { color:#060!important; }

#tab04 { position:relative; width:100px; margin:50px; }

#tab04 h3 { position:relative; z-index:1; height:16px; padding-top:4px; margin-bottom:-1px; border:solid #ccc; border-width:1px 0 1px 1px; text-align:center; font-family:宋体; background:#eee; cursor:pointer; }

#tab04 h3.up { z-index:3; color:#c00; background:#fff; }

#tab04 div.tab { display:none; position:absolute; left:99px; top:0; z-index:2; width:300px; height:200px; padding:5px; border:solid 1px #ccc; color:#666; }

#tab04 div.tab.up { display:block; }

-->

</style>

</head>

<body>

<div id="tab04">

<h3>首页</h3>

<div class="tab"><p>嘿嘿，无视h3原始class值。</p></div>

<h3>测试</h3>

<div class="tab wushi"><p>继续无视div原始class值。</p></div>

<h3>无聊</h3>

<div class="tab"><p>h3没有值也可以～</p></div>

<h3 class="box">傻蛋</h3>

<div class="tab tab123"><p>class值相似一样也可以～</p><div><p>指定class后，即时再多一个div也行。</p></div></div>

</div>

<script type="text/javascript">

function Pid(id,tag){

if(!tag){

return document.getElementById(id);

}

else{

return document.getElementById(id).getElementsByTagName(tag);

}

}

function tab(id,hx,box,iClass,s,pr){

var hxs=Pid(id,hx);

var boxs=Pid(id,box);

if(!iClass){ // 如果不指定class，则：

boxsClass=boxs; // 直接使用box作为容器

}

else{ // 如果指定class，则：

var boxsClass = [];

for(i=0;i<boxs.length;i++){

if(boxs[i].className.match(/\btab\b/)){// 判断容器的class匹配

boxsClass.push(boxs[i]);

}

}

}

if(!pr){ // 如果不指定预展开容器，则：

go\_to(0); // 默认展开序列

yy();

}

else {

go\_to(pr);

yy();

}

function yy(){

for(var i=0;i<hxs.length;i++){

hxs[i].temp=i;

if(!s){// 如果不指定事件，则：

s="onmouseover"; // 使用默认事件

hxs[i][s]=function(){

go\_to(this.temp);

}

}

else{

hxs[i][s]=function(){

go\_to(this.temp);

}

}

}

}

function go\_to(pr){

for(var i=0;i<hxs.length;i++){

if(!hxs[i].tmpClass){

hxs[i].tmpClass=hxs[i].className+=" ";

boxsClass[i].tmpClass=boxsClass[i].className+=" ";

}

if(pr==i){

hxs[i].className+=" up"; // 展开状态：标题

boxsClass[i].className+=" up"; // 展开状态：容器

}

else {

hxs[i].className=hxs[i].tmpClass;

boxsClass[i].className=boxsClass[i].tmpClass;

}

}

}

}

tab("tab1","h3","div","tab");

//-->

</script>

</body>

</html>

**横向选项卡**

1 首先引入jquery库

2 初始化事件

**[javascript]** [view plain](http://blog.csdn.net/yaozhiwei1019/article/details/53389642) [copy](http://blog.csdn.net/yaozhiwei1019/article/details/53389642)

* <script type="text/javascript">
* $(**function** () {
* tabs('#oauthTabs', 'mouseover');
* });
* </script>

可以绑定各种事件，比如click、mouseover等。

3 tabs方法

* <script>
* //Tab控制选项卡
* **function** tabs(tabObj, event) {
* //绑定事件
* **var** tabItem = $(tabObj).find(".tab-head ul li a");
* tabItem.bind(event, **function** () {
* //设置点击后的切换样式
* tabItem.removeClass("selected");
* $(**this**).addClass("selected");
* //设置点击后的切换内容
* **var** tabNum = tabItem.parent().index($(**this**).parent());
* $(tabObj).find(".tab-content").hide();
* $(tabObj).find(".tab-content").eq(tabNum).show();
* });
* }
* </script>

4 html代码

* **<div** id="oauthTabs"**>**
* <!--选项卡-->
* **<div** class="tab-head"**>**
* **<ul>**
* **<li>**
* **<a** class="selected" href="javascript:;"**>**绑定已有的账号**</a>**
* **</li>**
* **<li>**
* **<a** href="javascript:;"**>**创建新的账号**</a>**
* **</li>**
* **</ul>**
* **</div>**
* <!--/选项卡-->
* <!--选项内容-->
* **<div** class="tab-content" style="display: block;"**>**
* **<dl>**
* **<dt>**用户名：**</dt>**
* **<dd>**
* **<input** name="txtUserName" **/></dd>**
* **</dl>**
* **<dl>**
* **<dt>**密　码：**</dt>**
* **<dd>**
* **<input** name="txtPassword" type="password" **/></dd>**
* **</dl>**
* **<dl>**
* **<dt></dt>**
* **<dd>**
* **<input** id="btnSubmit1" name="btnSubmit1" type="submit" class="btn" value="立即绑定" **/></dd>**
* **</dl>**
* **</div>**
* **<div** class="tab-content"**>**
* **<dl>**
* **<dt>**手机号码：**</dt>**
* **<dd>**
* **<input** name="txtMobile" type="text" **/></dd>**
* **</dl>**
* **<dl>**
* **<dt>**登录密码：**</dt>**
* **<dd>**
* **<input** name="txtPassword" type="password" **/></dd>**
* **</dl>**
* **<dl>**
* **<dt></dt>**
* **<dd>**
* **<input** id="btnSubmit2" name="btnSubmit1" type="submit" class="btn" value="创建新账户" **/></dd>**
* **</dl>**
* **</div>**
* <!--/选项内容-->
* **</div>**

**Javascript前端验证码**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<!DOCTYPE html>

<html>

<head>

<title></title>

<style type="text/css">

div{

border: dashed;

width:200px;

height:100px;

font-size:35px;

}

</style>

<script src="js/jquery-1.9.1.min.js"></script>

<script type="text/javascript">

$(document).ready(function() {

//我写的验证码

//验证码

var code;

function createCode(){

code = '';//首先默认code为空字符串

var codeLength = 4;//设置长度，这里看需求，我这里设置了4

var codeV = $("div");

//设置随机字符

var random = new Array(0,1,2,3,4,5,6,7,8,9,'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R', 'S','T','U','V','W','X','Y','Z');

for(var i = 0; i < codeLength; i++){ //循环codeLength 我设置的4就是循环4次

var index = Math.floor(Math.random()\*36); //设置随机数范围,这设置为0 ~ 36

code += random[index]; //字符串拼接 将每次随机的字符 进行拼接

}

codeV.text(code);//将拼接好的字符串赋值给展示的Value

}

//页面开始加载验证码

createCode();

//验证码Div加载点击事件

$("div").bind('click',function() {

createCode();

});

//下面就是判断是否==的代码，无需解释

$("#b1").bind('click',function() {

var oValue = $("#in1").val().toUpperCase();

$("#l1").html("");

if(oValue ==""){

$("#l1").html("<font color='red'>请输入验证码</font>");

}else if(oValue != code){

$("#l1").html("<font color='red'>验证码不正确，请重新输入</font>");

oValue = "";

createCode();

}else{

$("#l1").html("<font color='blue'>验证码正确</font>");

}

});

});

</script>

</head>

<body>

<center>

<label >请输入验证码：</label><input type="text" id="in1" value="" placeholder="请输入验证码">

<button id="b1">点击验证</button>

<div></div><label id="l1"></label>

</center>

</body>

</html>

**账号 用户名+验证+验证码（后端提供）**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="Content-Type" content="text/html;charset=UTF-8" />

<title>农牧大家评-注册</title>

<link rel="stylesheet" href="\_\_PUBLIC\_\_/Css/regis.css" />

<script src="js/jquery.js"></script>

<script type="text/javascript">

//ajax请求地址

var checkAccount = "{:U('checkAccount')}";

var checkUname = "{:U('checkUname')}";

var checkVerify = "{:U('checkVerify')}";

</script>

</head>

<body>

<div id='logo'></div>

<div id='reg-wrap'>

<form action="{:U('runRegis')}" method='post' name='register'>

<fieldset>

<legend><span style="font-weight:bold;font-size:16px;">用户注册</span></legend>

<p>

<label for="account">注册账号：</label>

<input type="text" name='account' id='account' class='input' placeHolder="6~16个字符" /><span style="margin-left:5px;color:red" id="accountflag"></span>

</p>

<p>

<label for="pwd">密码：</label>

<input type="password" name='pwd' id='password' placeHolder="6~16个字符" class='input'/><span style="margin-left:5px;color:red" id="passwordflag"></span>

</p>

<style type="text/css">

.chushi{

width:68px;

height:18px;

line-height:18px;

background:#EEEEEE;

display:inline-block;

margin-top:3px;

text-align:center;

}

.power{

width:68px;

height:18px;

line-height:18px;

background:red;

display:inline-block;

margin-top:3px;

text-align:center;

}

</style>

<p>

<label for="pwded">密码强度：</label>

<span class="pwd-letter">

<span class="chushi">弱</span>

<span class="chushi">中</span>

<span class="chushi">强</span>

</span>

</p>

<p>

<label for="pwded">确认密码：</label>

<input type="password" name='pwded' id="repassword" class='input'/>

<span style="margin-left:5px;color:red" id="repasswordflag"></span>

</p>

<p>

<label for="uname">昵称：</label>

<input type="text" name='uname' id='uname' placeHolder="3~10个字符" class='input'/>

<span style="margin-left:5px;color:red" id="unameflag"></span>

</p>

<p>

<label for="verify">验证码：</label>

<input type="text" name='code' class='input' id='code'/>

<img src="{:U('verify')}" width='80' height='20' id='code-img'/>

<span style="margin-left:5px;color:red" id="codeflag"></span>

</p>

<p class='run'>

<input type="submit" value='马上注册' id='regis'/>

</p>

</fieldset>

</form>

</div>

</body>

<script>

$(function(){

//刷新验证码

function changeImg(){

$("#code-img").attr('src',function(i,v){

return v+'?'+Math.random();

});

}

//点击刷新验证码

$("#code-img").click(function(){

//验证码的更换

changeImg();

});

//设置错误和正确的样式

function flagError(account){

$('#'+account).removeClass('valid1');

$('#'+account+'flag').removeClass('valid2');

$('#'+account).addClass('error1');

$('#'+account+'flag').addClass('error2');

}

function flagValid(account){

$('#'+account).removeClass('error1');

$('#'+account+'flag').removeClass('error2');

$('#'+account).addClass('valid1');

$('#'+account+'flag').addClass('valid2');

}

//验证账户

//设置状态

var accountflag=false;

function checkAccountjs(oo){

var patten=/^\w{6,16}$/;

if(!patten.test(oo.value)){

accountflag=false;

flagError('account');

$('#accountflag').html("账号必须为6~16位只包含字母,数字,下划线");

return false;

}else{

//利用Ajax实现用户名查询是否存在

$.ajax({

url:checkAccount,

data:{'account':oo.value},

dataType:'json',

type:'post',

success:function(msg){

if(!msg){

flagError('account');

$('#accountflag').html("账户已被占用");

return false;

}else{

flagValid('account');

$('#accountflag').html('');

accountflag=true;

}

}

});

}

}

$('#account').blur(function(){

//查询用户名是否合法

checkAccountjs(this);

});

//密码验证

//设置状态

var passwordflag=false;

function checkPassword(oo){

var patt = /^\w{6,16}$/;

if(!patt.test(oo.value)){

flagError('password');

$('#passwordflag').html('密码必须为6~16位只包含字母,数字,下划线');

return false;

}else{

flagValid('password');

$('#passwordflag').html('');

passwordflag=true;

}

}

//判断密码强度

function qiangdu(oo){

var pwd\_letter=$(".pwd-letter span");

var modes=0;

var pwd=oo.value;

if (pwd.length < 6 || pwd.length > 16){

pwd\_letter.eq(modes).removeClass('power').siblings().removeClass('power');

pwd\_letter.eq(modes).addClass('chushi').siblings().addClass('chushi');

}else{

if (/\d/.test(pwd)) modes++; //数字

if (/[a-z]/.test(pwd)) modes++; //小写

if (/[A-Z\_]/.test(pwd)) modes++; //大写

pwd\_letter.eq(modes-1).removeClass('chushi').siblings().removeClass('power');

pwd\_letter.eq(modes-1).addClass('power').siblings().addClass('chushi');

}

}

$('#password').keyup(function(){

//查询用户名是否合法

checkPassword(this);

qiangdu(this);

});

//确认密码一致

//设置状态

var repasswordflag=false;

function checkRepassword(oo){

var password = $('#password').val();

if(password != oo.value){

if(oo.value.length>0){

flagError('repassword');

$('#repasswordflag').html('两次密码输入不一致');

return false;

}

}else{

if(oo.value.length>0){

flagValid('repassword');

$('#repasswordflag').html('');

repasswordflag=true;

}

}

}

$('#repassword').keyup(function(){

if (this.value.length < 6 || this.value.length > 16){

return false;

}

checkRepassword(this);

});

// //判断长度

// function strlen(str) {

// var len = 0;

// for (var i = 0; i < str.length; i++) {

// var c = str.charCodeAt(i);

// //单字节加1

// if ((c >= 0x0001 && c <= 0x007e) || (0xff60 <= c && c <= 0xff9f)) {

// len++;

// }

// else {

// len += 1;

// }

// }

// return len;

// }

//验证昵称

//设置状态

var unameflag=false;

function checkUnamejs(oo){

var patten=/^[\u4e00-\u9fa5\_\w]{3,10}$/;

if(!patten.test(oo.value)){

unameflag=false;

flagError('uname');

$('#unameflag').html("昵称必须为3~10位只包含字母,数字,下划线,中文");

return false;

}else{

//利用Ajax实现用户名查询是否存在

$.ajax({

url:checkUname,

data:{'uname':oo.value},

dataType:'json',

type:'post',

success:function(msg){

if(!msg){

flagError('uname');

$('#unameflag').html("账户已被占用");

return false;

}else{

flagValid('uname');

$('#unameflag').html('');

unameflag=true;

}

}

});

}

}

$('#uname').blur(function(){

//查询用户名是否合法

checkUnamejs(this);

});

//验证码

//设置状态

var codeflag=false;

function checkCode(oo){

if(oo.value.length!=4){

flagError('code');

$('#codeflag').html("验证码格式不对");

return false;

}

if (oo.value.length==4) {

$.ajax({

url:checkVerify,

data:{'code':oo.value},

dataType:'json',

type:'post',

success:function(msg){

if(!msg){

flagError('code');

$('#codeflag').html("验证码错误,请重新输入");

return false;

}else{

flagValid('code');

$('#codeflag').html('');

codeflag=true;

}

}

});

}

}

$('#code').keyup(function(){

checkCode(this);

});

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

if (accountflag&&passwordflag&&repasswordflag&&unameflag&&codeflag) {

return true;

}else{

checkAccountjs($('#account')[0]);

checkPassword($('#password')[0]);

qiangdu($('#password')[0]);

checkRepassword($('#repassword')[0]);

checkUnamejs($('#uname')[0]);

checkCode($('#code')[0]);

return accountflag&&passwordflag&&repasswordflag&&unameflag&&codeflag;

}

return false;

});

});

</script>

<!--

<?php

namespace Home\Controller;

use Think\Controller;

/\*\*

\* 注册与登录控制器

\*/

Class AccessController extends Controller {

/\*\*

\* 登录页面

\*/

Public function index () {

$this->display();

}

/\*\*

\* 注册页面

\*/

Public function register () {

$this->display();

}

/\*\*

\* 注册表单处理

\*/

Public function runRegis () {

//post提交

if (!IS\_POST) {

$this->error('页面不存在');

}

//验证码判断

$code = $\_POST['code'];

$verify = new \Think\Verify();

$verify->reset=false;

if (!$verify->check($code)) {

$this->error('验证码错误');

}

//验证账户

$patten='/^\w{6,16}$/';

if(!preg\_match($patten, $\_POST['account'])){

$this->error('账户格式不对');

}

$where = array('account' => $account);

if (D('user')->where($where)->find()) {

$this->error('账户已被占用');

}

//验证密码格式

$patten='/^\w{6,16}$/';

if(!preg\_match($patten, $\_POST['pwd'])){

$this->error('密码格式不对');

}

if ($\_POST['pwd'] != $\_POST['pwded']) {

$this->error('两次密码不一致');

}

//验证昵称

$patten='/^[\x{4e00}-\x{9fa5}\w]{3,10}$/u';

echo $\_POST['uname'];

if(!preg\_match($patten, $\_POST['uname'])){

$this->error('昵称格式不对');

}

$uname=$\_POST['uname'];

$where = array('uname' => $uname);

if (D('userinfo')->where($where)->find()) {

$this->error('昵称已被占用');

}

//提取POST数据

$data = array(

'account' =>I('post.account'),

'password' =>md5(I('post.pwd')) ,

'registime' => $\_SERVER['REQUEST\_TIME'],

'userinfo' => array(

'username' => I('post.uname')

)

);

$id = D('UserRelation')->insert($data);

if ($id) {//插入数据成功后把用户ID写SESSION

session('uid', $id);

session('uname', $uname);

//跳转至首页

$this->success('注册成功，正在为您跳转...','Home/Index/index');

} else {

$this->error('注册失败，请重试...');

}

}

/\*\*

\* 获取验证码

\*/

Public function verify() {

$Verify = new \Think\Verify();

$Verify->reset = true;

$Verify->length = 4;

$Verify->useNoise = false;

$Verify->entry();

}

/\*\*

\* 异步验证账号是否已存在

\*/

Public function checkAccount () {

if (!IS\_POST) {

$this->error('页面不存在');

}

$account =I('post.account');

$where = array('account' => $account);

if (D('user')->where($where)->find()) {

echo 'false';

} else {

echo 'true';

}

}

/\*\*

\* 异步验证昵称是否已存在

\*/

Public function checkUname () {

if (!IS\_POST) {

$this->error('页面不存在');

}

$username =I('post.uname');

$where = array('username' => $username);

if (D('userinfo')->where($where)->find()) {

echo 'false';

} else {

echo 'true';

}

}

/\*\*

\* 异步验证验证码

\*/

Public function checkVerify () {

if (!IS\_POST) {

$this->error('页面不存在');

}

$code = I('post.code');

$verify = new \Think\Verify();

$verify->reset=false;

if (!$verify->check($code)) {

echo 'false';

} else {

echo 'true';

}

}

}

?>

-->

</html>

**Jquery form js提交表单**

<script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>  
<script src="http://malsup.github.io/jquery.form.js"></script>

<form>  
    标题：<input type="text" name="title" /><br />  
    内容：<textarea name="content"><textarea/><br />  
    <button>提交</button>  
</form>

$('button').on('click', function() {  
  
    $('form').on('submit', function() {  
        var title = $('inpur[name=title]').val(),  
            content = $('textarea').val();  
  
        $(this).ajaxSubmit({  
            type: 'post', // 提交方式 get/post  
            url: 'your url', // 需要提交的 url  
            data: {  
                'title': title,  
                'content': content  
            },  
            success: function(data) { // data 保存提交后返回的数据，一般为 json 数据  
                // 此处可对 data 作相关处理  
                alert('提交成功！');  
            }  
            $(this).resetForm(); // 提交后重置表单  
        });  
        return false; // 阻止表单自动提交事件  
    });  
});

**jquery邮箱验证**

<input type="email" name="email" id="email" value="" onpaste="return false"/>

Js

//验证邮箱

function vailEmail(){

var email = jQuery("#email").val();

var flag = false;

var message = "";

var myreg = /^([\.a-zA-Z0-9\_-])+@([a-zA-Z0-9\_-])+(\.[a-zA-Z0-9\_-])+/;

if(email ==''){

message = "邮箱不能为空！";

}else if(!myreg.test(email)){

message = "请输入有效的邮箱地址！";

}else if(checkEmailIsExist()){

message = "该邮箱地址已经被注册！";

}else{

flag = true;

}

if(!flag){  
　　　　　　　　　　//错误提示

//jQuery("#emailDiv").removeClass().addClass("ui-form-item has-error");

// jQuery("#emailP").html("");

//jQuery("#emailP").html("<i class=\"icon-error ui-margin-right10\">&nbsp;<\/i>"+message);

//jQuery("#email").focus();

}else{  
 //正确提示

//jQuery("#emailDiv").removeClass().addClass("ui-form-item has-success");

//jQuery("#emailP").html("");

//jQuery("#emailP").html("<i class=\"icon-success ui-margin-right10\">&nbsp;<\/i>该邮箱可用");

}

return flag;

}



写一个方法来验证一下



//验证邮箱是否存在

function checkEmailIsExist(){

var email = jQuery("#email").val();

var flag = false;

jQuery.ajax(

{ url: "checkEmail?t=" + (new Date()).getTime(),

data:{email:email},

dataType:"json",

type:"GET",

async:false,

success:function(data) {

var status = data.status;

if(status == "1"){

flag = true;

}

}

});

return flag;

}

# Selec选择某个data弹出input

<script>

function showText(){

if(document.getElementById("q").value==1){

document.getElementById("text").innerHTML="<input name='a' id='a' value='第一个文本框' type='text' /><input type='button' value='隐藏' onclick='hiddenText();' />";

document.getElementById("q").value=1;

}else if(document.getElementById("q").value==2){

document.getElementById("text").innerHTML="<input name='b' id='b' value='第二个文本框' type='text' /><input type='button' value='隐藏' onclick='hiddenText();' />";

document.getElementById("q").value=2;

}

}

function hiddenText(){

document.getElementById("text").innerHTML="";

}

</script>

<select name="q" id="q" onchange="showText();">

<option value="1">1</option>

<option value="2">2</option>

</select>

<span id="text"></span>

# 无刷新自动更新某个数值（论询）

/\*第一次读取最新通知\*/

setTimeout(function() {

Push();

},

200);

/\*30轮询读取函数\*/

setInterval(function() {

Push();

},

30000);

/\*请求函数的ajax\*/

function Push() {

$.ajax({

type: "POST",

url: "/index.php?s=Push&a=index",

data: {

t: 3

},

beforeSend: function() {},

success: function(data) {

var obj = eval("(" + data + ")");

// alert(obj.sixin);

if (obj.sixin != 0) {

$(".tongzhi").html(obj.sixin).show();

} else {

$(".tongzhi").html(0).hide();

}

}

# 修改后自动ajax提交数据

<table>

<tr>

<td class="canChange">sd</td>

</tr>

<input type="text" id="username" />

</table>

</body>

<script>

//JS

$(document).ready(function(){

td\_Click();

})

//点击事件

function td\_Click() {

$(".canChange").click(function () {

var td = $(this);

//所点文本框的id

var id = $(this).attr("id")

var txt = $.trim(td.text());

var input = $("<input class=\"myinput\" id='username' name='username' type='text'value='" + txt + "'style=\"width:80%;heigth:100%;\"/>");

td.html(input);

input.click(function () { return false; });

//获取焦点

input.trigger("focus");

//文本框失去焦点后提交内容，重新变为文本

input.blur(function () {

// var newtxt = $(this).val();

var newtxt = $("#username").val();

//判断文本有没有修改

if (newtxt != txt) {

if (newtxt == null || newtxt == "") {

td.html(txt);

}

else {

//表示已经修改

$.post("url",{a: 1, b: 2},function(data){

$(".canChange").val(result);

})

}

}

else {

td.html(txt);

}

});

});

}

</script>

# Ajax更新某个数据

<script>

/\*第一次读取最新通知\*/

setTimeout(function() {

Push();

},

200);

/\*30轮询读取函数\*/

setInterval(function() {

Push();

},

30000);

/\*请求函数的ajax\*/

function Push(id) {

$.ajax({

type: "post",

url: "",

data: {

'id': id

},

async: true,

dataType: 'json',

success: function(data) {

var goods = '';

$.each(data, function(k, v) {

goods+='';

});

$('tbody').html(goods);

}

});

}

});

</script>