表单验证

1. 表单验证的目的
2. 减轻服务器的压力
3. 保证用户输入数据的合法性
4. 实现表单验证的思路:
5. 制作规范的表单

|  |
| --- |
| <form id="form1" name="form1" method="post" action="处理页面:server.html">  <p>用户名:  <input type="text" name="textfield" id="textfield" />  6-12位  </p>  <p>  <input type="submit" name="button" id="button" value="提交" />  </p>  </form>  注意:当点击提交按钮时，表单会自动跳转到form标签的action属性所指定的页面进行处理. |

1. 利用表单的onsubmit事件实现表单验证

|  |
| --- |
| <form id="form1" name="form1" onsubmit="return false;" method="post" action="server.html">  注意:onsubmit事件接收返回值，如果返回true则提交表单、如果返回false则不提交表单. |

表单验证的辅助事件:

Onblur 失去焦点事件

1. 验证表音时，有多个表单对象进行验证

|  |
| --- |
| <script language="javascript">  //一个表单对象一个验证的方法，提交按钮验证所有  //验证用户名  function checkUname(){  //获取用户名  var uname=$("#uname").val();  var len=uname.length;  if(len>=6&&len<=12){  $("#noticeUname").text("OK").css("color","green");  return true;  }else{  $("#noticeUname").text("长度为6-12位").css("color","red");  //将光标定位在文本框  //$("#uname")[0].focus(); //获得焦点  $("#uname")[0].select(); //获得选中内容  return false;  }  }    /\*检查邮箱\*/  function checkEmail(){  //取值  var email=$("#email").val();  var pos=email.indexOf("@");  var pos1=email.lastIndexOf(".");  if(pos!=-1 && pos1!=-1 && (pos1-pos)>1){  $("#noticeEmail").text("OK").css("color","green");  return true;  }else  {  $("#noticeEmail").text("邮箱格式不正确").css("color","red");  $("#email")[0].select(); //获得选中内容  return false;  }  }  /\*表单提交检查所有\*/  function checkAll(){  if(checkUname()&&checkEmail())  return true;  else  return false;  }  </script>  HTML代码:  <form id="form1" name="form1" onsubmit="return checkAll();" method="post" action="server.html">  <p>用户名:  <input type="text" name="textfield" onblur="checkUname();" id="uname" /><span id="noticeUname"></span>  6-12位  </p>  <p>邮箱:  <input onblur="checkEmail();" type="text" name="email" id="email" /><span id="noticeEmail"></span>  </p>  <p>  <input type="submit" name="button" id="button" value="提交" />  </p>  </form> |

1. 验证的方法

方式一:采用字符串处理验证表单

方式二:采用正则表达式验证

1. 正则表达式
2. 定义正则表达式

var 对象名称=/^正则表达式$/附加参数

注意:^表示正则表达式开始 $表示正则表达式结束

或者

Var对象名称=new RegExp(“正则表达式”[,附加参数]);

1. (了解)正则表达式的书写

名词形容词名词形容词

名词:

[列举可选字符]

\d 表示0-9的数字 [0-9]

\D 表示非0-9的数字 [^0-9]

\w 表示一个数字、下划线、字母 [A-Za-z0-9\_]

\W 表示非一个数字、下划线、字母

. 表示除换行以外所有字符

形容词:

{n} 表示匹配n次

{n,} 表示匹配至少n次

{n,m} 表示匹配n到m次

\* 表示匹配0-无穷次

+ 表示匹配1-远穷次

？ 表示可有可无

3. 使用正则表达式的test方法匹配内容,匹配成功true,反之false

|  |
| --- |
| //验证电话号码  function checkTel(){  var tel=$("#tel").val(); //输入内容  //定义正则表达式  var reg=/^((0?13[035]\d{8})|(\d{8}))$/; //正确格式    //使用正则表达式的test方法匹配内容,匹配成功true,反之false  if(reg.test(tel)){  $("#noticeTel").text("OK").css("color","green");  }  else{  $("#noticeTel").text("格式不正确").css("color","red");  }    } |