|  |
| --- |
| 佛山大学  实验报告  实验名称 ASP.NET验证控件的使用  实验项目 验证控件的使用  专业班级 22物联网工程2班 姓名 学号  指导教师 罗 平 成绩 日期 2024.10.18     1. 实验目的   1、了解常用验证控件。  2、掌握验证控件的使用。   1. 实验内容 2. 非空验证RequiredFieldValidator控件、限定输入范围RangeValidator控件、比较验证CompareValidator控件、自定义验证CustomValidator控件。 3. 显示验证信息ValidationSummary控件。 4. 实验步骤 5. RequiredFieldValidator非空验证控件使用。   1.页面  <body>      <form id="form1" runat="server">      <div>            <asp:TextBox ID="InputBox" runat="server"></asp:TextBox>          <asp:RequiredFieldValidator ID="rfInputBox" runat="server"              ErrorMessage="输入不能为空！" ControlToValidate="InputBox" Font-Bold="True"></asp:RequiredFieldValidator>          <br />          <asp:Button ID="ValidateBtn" runat="server" Text="测试验证"              onclick="ValidateBtn\_Click" />          <br />          <asp:Label ID="ValidateMsg" runat="server" ></asp:Label>        </div>      </form>  </body>   1. 代码   public partial class \_Default : System.Web.UI.Page  {      protected void Page\_Load(object sender, EventArgs e)      {          ValidateMsg.Visible = false;      }      protected void ValidateBtn\_Click(object sender, EventArgs e)      {          if(Page.IsValid)          {              ValidateMsg.Text = "用户输入验证合法！";              ValidateMsg.Visible = true;              }              else              {                  ValidateMsg.Text = "用户输入验证失败！";                  ValidateMsg.Visible = true;                  }      }  }   1. RangeValidator限定输入范围验证控件使用。   1.页面  <body>      <form id="form1" runat="server">      <asp:Label ID="Label2" runat="server" Text="范围验证："></asp:Label>      <br />      <asp:TextBox ID="InputBox" runat="server"></asp:TextBox>      <asp:RangeValidator ID="rvlnput" runat="server"          ErrorMessage="只能输入在0~100之间"  ControlToValidate="InputBox" Font-Bold="True"          MaximumValue="100" MinimumValue="0" Type="Integer"></asp:RangeValidator>      <div>            <asp:TextBox ID="InputBoxLetter" runat="server"></asp:TextBox>          <asp:RangeValidator ID="RangeValidator1" runat="server"              ErrorMessage="只能输入A~Z之间" ControlToValidate="InputBoxLetter"              Font-Bold="True" MaximumValue="Z" MinimumValue="A"></asp:RangeValidator>          <br />          <asp:Button ID="Button1" runat="server" Text="验证" onclick="Button1\_Click" />          <br />          <asp:Label ID="ValidateMsg" runat="server"></asp:Label>        </div>      </form>  </body>  2.代码  public partial class RangeValidator : System.Web.UI.Page  {      protected void Page\_Load(object sender, EventArgs e)      {          ValidateMsg.Visible = false;      }      protected void Button1\_Click(object sender, EventArgs e)      {          if (Page.IsValid)          {              ValidateMsg.Text = "用户输入验证合法！";              ValidateMsg.Visible = true;          }          else          {              ValidateMsg.Text = "用户输入验证失败！";              ValidateMsg.Visible = true;          }      }  }   1. CompareValidator比较验证控件使用。   1.页面  <body>      <form id="form1" runat="server">      <div>            <table style="width:100%;" border="0" cellpadding="5" cellspacing="5"              width="100%">              <tr>                  <td class="style1">                      <asp:Label ID="Label1" runat="server" Text="对比验证" Font-Bold="True"></asp:Label>                  </td>                </tr>              <tr>                  <td>                      <asp:TextBox ID="InputBox" runat="server"></asp:TextBox>                  </td>                  </tr>              <tr>                  <td>                      <asp:TextBox ID="InputBoxLetter" runat="server"></asp:TextBox>                      <asp:CompareValidator ID="cvlnput" runat="server" ControlToCompare="InputBox"                          ControlToValidate="InputBoxLetter" ErrorMessage="两次密码不一致" Font-Bold="True"></asp:CompareValidator>                      <br />                      <asp:Button ID="Button1" runat="server" Text="验证" onclick="Button1\_Click" />                      <br />          <asp:Label ID="ValidateMsg" runat="server"></asp:Label>                    </td>            </table>        </div>      </form>  </body>  2.代码  public partial class Compare : System.Web.UI.Page  {      protected void Page\_Load(object sender, EventArgs e)      {          ValidateMsg.Visible = false;      }      protected void Button1\_Click(object sender, EventArgs e)      {          if (Page.IsValid)          {              ValidateMsg.Text = "用户输入验证合法！";              ValidateMsg.Visible = true;          }          else          {              ValidateMsg.Text = "用户输入验证失败！";              ValidateMsg.Visible = true;          }      }  }   1. CustomValidator自定义验证控件使用。   1.页面  <body>      <script language="javascript">          function DivThreeValidate(source, argument)        {             if(argument.Value % 3)  　　　  {             argument.IsVaild = false;           }             else           {  　　　　   argument.IsValid = true;  　　　   }  　　　}      </script>      <form id="form1" runat="server">      <div>          <table style="width: 100%;">              <tr>                  <td class="style1">                      <asp:Label ID="Label1" runat="server" Font-Bold="True" Text="自定义验证："></asp:Label>                  </td>              </tr>              <tr>                  <td class="style3">                      &nbsp;<asp:TextBox ID="inputBox" runat="server"></asp:TextBox>                      <asp:CustomValidator ID="cvlnput" runat="server" ErrorMessage="输入信息不是偶数！"                          onservervalidate="cvlnput\_ServerValidate" ControlToValidate="inputBox"                          Font-Bold="True"></asp:CustomValidator>                  </td>              </tr>              <tr>                  <td class="style3">                      &nbsp;<asp:TextBox ID="InputBoxLetter" runat="server"></asp:TextBox>                      <asp:CustomValidator ID="cvNumber" runat="server"                          ErrorMessage="输入数字不能整除3！" ClientValidationFunction="DivThreeValidate"                          ControlToValidate="InputBoxLetter" Font-Bold="True"                          onservervalidate="cvNumber\_ServerValidate"></asp:CustomValidator>                  </td>                  </tr>              <tr>                  <td class="style3">                      <asp:Button ID="ValidateBtn" runat="server" Text="验证"                          onclick="ValidateBtn\_Click" />                  </td>                  </tr>              <tr>                  <td class="style2">          <asp:Label ID="ValidateMsg" runat="server"></asp:Label>                    </td>              </tr>          </table>          <br />      </div>      </form>  </body>  2.代码  public partial class Custom : System.Web.UI.Page  {      protected void Page\_Load(object sender, EventArgs e)      {          ValidateMsg.Visible = false;      }        protected void ValidateBtn\_Click(object sender, EventArgs e)      {          if (Page.IsValid)          {              ValidateMsg.Text = "用户输入验证合法！";              ValidateMsg.Visible = true;          }          else          {              ValidateMsg.Text = "用户输入验证失败！";              ValidateMsg.Visible = true;          }      }      protected void cvlnput\_ServerValidate(object source, ServerValidateEventArgs args)      {          args.IsValid = false;          try          {              int number = Int32.Parse(args.Value);              if (number % 2 == 0)              {                  args.IsValid = true;              }          }          catch (Exception ex)          {              Response.Write("错误");          }      }      protected void cvNumber\_ServerValidate(object source, ServerValidateEventArgs args)      {          {              args.IsValid = false;              try              {                  int number = 0;                  if (Int32.TryParse(args.Value, out number) == true)                  {                      if (number % 3 == 0)                      {                          args.IsValid = true;                      }                  }              }              catch (Exception ex)              {                  Response.Write("错误");              }          }      }  }   1. ValidationSummary显示验证信息控件使用。   1.页面  <body>      <form id="form1" runat="server">      <div>            <table style="width:100%;">              <tr>                  <td class="style1">                      <asp:Label ID="Label1" runat="server" Font-Bold="True" Text="页面统一验证："></asp:Label>                  </td>              </tr>              <tr>                  <td>                      <asp:TextBox ID="InputBox" runat="server"></asp:TextBox>                      <asp:RequiredFieldValidator ID="rfInput" runat="server"                          ControlToValidate="InputBox" Display="None" ErrorMessage="输入不能为空！"                          Font-Bold="True"></asp:RequiredFieldValidator>                      <asp:RangeValidator ID="rvInput" runat="server" ControlToValidate="InputBox"                          Display="None" ErrorMessage="只能输入在0～100之间！" Font-Bold="True" MaximumValue="100"                          MinimumValue="0" Type="Integer"></asp:RangeValidator>                  </td>              </tr>              <tr>                  <td class="style1">                      <asp:TextBox ID="Email" runat="server"></asp:TextBox>                      <asp:RequiredFieldValidator ID="rfEmail" runat="server"                          ControlToValidate="Email" Display="None" ErrorMessage="输入不能为空！"                          Font-Bold="True"></asp:RequiredFieldValidator>                      <asp:RegularExpressionValidator ID="reEmail" runat="server"                          ControlToValidate="Email" Display="None" ErrorMessage="Email的格式不正确！"                          Font-Bold="True"                          ValidationExpression="\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*"></asp:RegularExpressionValidator>                  </td>              </tr>              <tr>                  <td class="style1">                      <asp:TextBox ID="Phone" runat="server"></asp:TextBox>                      <asp:RequiredFieldValidator ID="rfPhone" runat="server"                          ControlToValidate="Phone" Display="None" ErrorMessage="输入不能为空！"                          Font-Bold="True"></asp:RequiredFieldValidator>                      <asp:RegularExpressionValidator ID="reInput" runat="server"                          ControlToValidate="Phone" Display="None" ErrorMessage="手机号码格式不正确！"                          Font-Bold="True" ValidationExpression="(\d{2,3}-){0,1}\d{11}"></asp:RegularExpressionValidator>                  </td>              </tr>              <tr>                  <td class="style1">                      <asp:ValidationSummary ID="vsPage" runat="server" />                  </td>              </tr>              <tr>                  <td>          <asp:Button ID="Button1" runat="server" Text="验证" onclick="Button1\_Click" />          <br />          <asp:Label ID="ValidateMsg" runat="server"></asp:Label>                    </td>              </tr>          </table>        </div>      </form>  </body>  2.代码  public partial class ValidatorSummary : System.Web.UI.Page  {      protected void Page\_Load(object sender, EventArgs e)      {          ValidateMsg.Visible = false;      }      protected void Button1\_Click(object sender, EventArgs e)      {          if (Page.IsValid)          {              ValidateMsg.Text = "用户输入验证合法！";              ValidateMsg.Visible = true;          }          else          {              ValidateMsg.Text = "用户输入验证失败！";              ValidateMsg.Visible = true;          }      }    }   1. 网站截图            1. 实验调试及讨论   本次实验主要对几个验证方法进行实践，本质上说他们都大差不错，区别只是对比方式不同，实验中几个控件的使用也是挺有趣的。   1. 软件清单   Microsoft Visual Studio2010 |