

十天 学会 ASP.NET

- 第一天：调试出安装环境
- 第二天：掌握C#的基本知识
- 第三天：掌握WEB控件的使用(一)
- 第四天：掌握WEB控件的使用(二)
- 第五天：学会连接两种数据库
- 第六天：学会读取数据库
- 第七天：学会添加删除修改记录
- 第八天：初步掌握DATAGRID的使用
- 第九天：学会用DATAGRID分页显示
- 第十天：学会SESSION和COOKIE的使用
-
-
-
-

十天学会ASP.NET第一天

我认为ASP.NET比ASP难很多，希望大家做好准备。难难在实现同一个效果有很多方法，大家或许会疑惑。十天学会系列教程会清晰的写出各种方法，让大家能少一点坎坷。在ASP里面，大家几乎都是使用VB，数据库嘛，也无非是SQL和ACCESS。在ASP.NET里面，数据库还是这两个但是常用的语言多了一个C#，也是微软推荐的语言，为.NET平台定制的语言。这样ASP.NET就有四种组合：VB+ACCESS，VB+SQL，C#+ACCESS，C#+SQL，本教程会使用C#+ACCESS，兼顾四种来写，让大家对ASP.NET有一个全面的认识。虽然说学习ASP.NET不需要任何ASP基础，但是我觉得如果大家ASP不会，还是先看一下【十天学会ASP教程】，大家所需要了解的不是ASP的程序怎么写，而是怎么构建服务器，怎么使用HTML表单，同时对SQL语句有一个基础和理解，因为在本文里我不会将一些基础的东西。

废话不多说，开始我们第一天的学习：

学习目的：调试出安装环境

我要申明一点，操作系统就用2000，如果说要学习一个网络语言还不肯多装一个操作系统的话，那我看也不要学习了。先到网上下载一个MDAC(要是2.7以上的版本才行)，然后安装，安装以后或许会重新启动一次。然后再下载一个.NET FRAMEWORK SDK，安装以后会在控制面板里面的管理工具里面多两个东西，这个不用理睬他，我们要使用的还是原来的INTERNET服务管理器。打开它按照ASP教程里面的调试方法建立一个目录，建立一个EXAMPLE1.ASPX，然后用记事本打开这个文件在里面输入：

```
C# <script language="c#" runat="server">
void Page_Load()
{
    Response.Write("Hello World!");
}
</script>
```

```
VB <script language="vb" runat="server">
sub page_load()
response.write("Hello World!")
end sub
</script>
```

在这里我要说明两点：1、我的示例文件总是有A和B分别是用C#和VB写的，演示的图片就用C#那一种的，都一样嘛，教程里面代码也是写两种用<hr>分割开，大家可以比较一下。2、我写教程的时候用的都是记事本来编写APS.NET大家也可以安装VS.NET来编写不过起步还是用记事本比较好，因为可以看的更加透彻一点。

十天学会ASP.NET第二天

学习目的：掌握C#的基本知识

一、定义变量：

string strExample; 字符串("aaa")

char chrExample; 字符('a')

bool blnExample; Bool型(true/false)

DateTime datExample; 日期型("09/19/2002")

int intExample; 整数(32位有符号整数)

double dblExample; 浮点数(64位双精度浮点数)

二、各种运算：

= 赋值运算

+ 加法运算

- 减法运算

* 乘法运算

/ 除法运算

% 取模运算

&& 逻辑And

|| 逻辑Or

! 逻辑Not

三、各种结构：

```
if(条件)
```

```
{  
}
```

```
else
```

```
{  
}
```

```
switch(条件)
```

```
{
```

```
case option1:
```

```
break;
```

```
case option2:
```

```
break;
```

```
}
```

```
for(int i=1;i<=10;i++)//特别注意这里面是分号
```

```
{  
}
```

```
while(条件)
```

```
{  
}
```

```
do
```

```
{
```

```
}while(条件);
```

四、注意事项：

1、用习惯VB的人很容易漏了语句末尾的分号；

2、用习惯VB的人很随便使用变量但是不申明，这在C#默认情况下是不允许的；

3、在向函数传递参数的时候要用圆括号：Response.Write "aa";是不允许的。

我说的这点几乎称不上是在说一种语言，一种语言的学习不仅仅需要学习语法，还需要学习语言特色，希望大家有空还是买一本C#的书来好好研究一下C#作为一种新的语言的特性吧。

今天就说到这里，明天说一下WEB控件。

十天学会ASP.NET第三天

学习目的：掌握WEB控件的使用(一)

或许你会问，为什么需要WEB控件？WEB控件就好像VB里面的控件，可以在程序的任何地方引用，修改所有属性，有了它的配合才能完成代码分离。想一下，如果还是以前的HTML控件，我们当我们需要在某一个地方显示一段数据库的内容是不是只有用<%= %>这样势必在HTML包含了程序的代码块，就达不到分离代码和界面的目的了。

或许你还会问，为什么要代码分离那？原因很简单，不是每一个程序员都是合格的网页设计师，如果我们需要修改代码的时候会破坏网页设计师原来的设定，是不是不方便那，程序代码和HTML代码混合在一起对于程序员来说也不方便修改代码。当然，我们的网页设计师也需要学习一点WEB控件的知识。

下面来一个一个介绍：

一、Label控件：

主要属性：

Text 标签所显示的文本

范例：<asp:Label id="lblMessage" Text="aaa" runat="server"/>

需要注意的是我们不能遗漏runat="server"字样，还有所有的WEB控件都要包含在<form runat="server"></form>中间

下面看一个完整的例子，在程序里面修改Label的Text属性，以此改变网页的显示。

```
C# <script runat="server" language="c#">
void Page_Load()
{
    lblMessage.Text="Hello World!";
}
</script>
<html>
<head> <title>Label.aspx</title> </head>
<body>
<form runat="server">
<asp:Label id="lblMessage" runat="server"/>
</form>
</body>
</html>
```

```
VB <script runat="server" language="vb">
sub Page_Load
    lblMessage.Text="Hello World!"
end sub
</script>
<html>
<head> <title>Label.aspx</title> </head>
<body>
<form runat="server">
<asp:Label id="lblMessage" runat="server"/>
</form>
</body>
</html>
```

二、TextBox控件：

主要属性：

Text 控件显示的文本；

MaxLength 文本框可以添加的最多的字符数（多行文本框无效）；

ReadOnly 只读；

TextMode 有下面几个有效值 MultiLine>Password,SingleLine；

Rows 指定文本框的垂直尺寸

三、Button控件：

有三种：Button 标准的表单按钮；ImageButton 显示图像的表单按钮；LinkButton 显示作为超链接样子Button

Button基本上就是Text属性来设置按钮上面的文字；

ImageButton基本上就是ImageUrl来设置按钮上图象的地址；AlternativeText 在浏览器不支持图片的时候显示的文字；

LinkButton基本上就是Text属性来设置按钮上的文字；

他们三者共同的属性就是CausesValidation=true/false来设置按钮提交的表单是不是被检验（后面将说到检验控件）

他们三者共同的方法就是OnClick就是点击按钮的时候触发的函数，下面举一个例子：

C#

```
<script runat="server" language="c#">
void btnCounter_OnClick(object sender,EventArgs e)
{
    btnCounter.Text="Clicked";
}
</script>
<html>
<head> <title>Label.aspx</title> </head>
<body>
<form runat="server">
<asp:Button Text="UnClicked" OnClick="btnCounter_OnClick" id="btnCounter" runat="server"/>
</form>
</body>
</html>
```

VB

```
<script runat="server" language="vb">
sub btn_COunter_OnClick(s as object,e as eventargs)
    btnCounter.Text="Clicked"
end sub
</script>
<html>
<head> <title>Label.aspx</title> </head>
<body>
<form runat="server">
<asp:Button Text="UnClicked" OnClick="btnCounter_OnClick" id="btnCounter" runat="server"/>
</form>
</body>
</html>
```

今天就说这三个控件，明天继续。

十天学会ASP.NET第四天

学习目的：掌握WEB控件的使用(二)

首先我要说明，我这十天教程也只是一个让大家对ASP.NET有一个认识而已，我也跳过了很多东西。网上有很多人在责骂我写的教程，认为十天什么都学不会，看了我的教程还是什么都不会，我想他们或许是过高估计了我的教程。外面有很多书，关于ASP和ASP.NET的，我就是看外面的书学会的。我所要做的就是希望大家能看了我的教程以后觉得ASP和ASP.NET不是学不会的东西然后有信心去学会它。我想学习还是靠自己的，外面每一本书也有自己写的不周到的地方，关键还需要自己的实践加上探索。

首先来说一下表单的验证控件，以前我们做表单验证，如果是客户端的JAVASCRIPT，不能做到美观统一，而且还可以通过直接在浏览器起教表单来屏蔽；如果我们做服务器端验证，又会发现让检验结果在前台反映也是很麻烦的事情。现在好了，ASP.NET提供的很多检验控件让我们不需要考虑是客户端检验还是服务器端检验，让我们不需要考虑是怎么输出检验结果。

看一下检验表单控件是不是填写了的检验控件先：

```
<asp:textbox id="username" runat="server" cssclass="textbox"/>
<asp:requiredfieldvalidator controltovalidate="username" display="dynamic" forecolor="#ff0000" font-name="宋体"
font-size="9pt" text="请填写" runat="server"/>
```

controltovalidate就是你想检验的控件的ID；默认情况下不显示错误消息，地方也会被保留，如果使用了display="dynamic" 那么不显示错误消息的地方不会被空出；forecolor="#ff0000" font-name="宋体" font-size="9pt" 就是设定错误消息字体的颜色，字体，大小的；text="请填写" 就是当没有填写内容时候显示的错误消息；别忘记最后的runat="server"。现在这个检验控件是放在了textbox的后面，那么错误消息也在textbox后面显示，也可以放在其他地方。

再看一下比较表单控件的数据与另外一个数据的检验控件：

```
<asp:textbox id="password1" runat="server" textmode="password" cssclass="textbox"/>
<asp:requiredfieldvalidator controltovalidate="password1" display="dynamic" forecolor="#ff0000" font-name="宋体"
font-size="9pt" text="请填写" runat="server"/>
```

这里是第一个需要用户填写的密码框

我们经常会再设立一个密码框让用户再填写一次：

```
<asp:textbox id="password2" runat="server" textmode="password" cssclass="textbox"/>
<asp:requiredfieldvalidator controltovalidate="password2" display="dynamic" forecolor="#ff0000" font-name="宋体"
font-size="9pt" text="请填写" runat="server"/>
<asp:comparevalidator controltovalidate="password2" controltocompare="password1" display="dynamic"
operator="equal" forecolor="#ff0000" font-name="宋体" font-size="9pt" text="确认失败" runat="server"/>
```

controltocompare="password1" 就是需要比较的控件；operator="equal" 就是设定比较操作是：是否相等（还有NotEqual：不相等,LessThan：少于,GreaterThan：大于，当不符合这个比较操作的时候显示错误消息）。

以前说到BUTTON控件CausesValidation=true/false来设置按钮提交的表单是不是被检验也就是这里的几个检验控件，如果CausesValidation=true的话按下按钮就会首先检测是不是都通过了检验控件的检验。

下面再说两点：

一、每一个可以显示的控件都有是不是显示的属性，比如username.Visible=false;就是隐藏了前面的username控件；还有就是如果要对控件应用CSS的话就这么写：cssclass=""，而不是以前HTML中的class=""。

二、所有WEB控件都要包含在<form runat="server"></form>中，否则就会出错。

今天就到此结束，请继续往后面看，下次开始就要说ADO.NET了。

十天学会ASP.NET第五天

学习目的：学会连接两种数据库

对于ASP来说，我们常用的数据库无非是ACCESS和SQL SERVER，对于ASP.NET也是，不过ASP.NET对于SQL SERVER有专门的连接组件而不推荐用OLE DB。

首先看一下ACCESS的连接数据库并打开；

```
tring strConnection="Provider=Microsoft.Jet.OleDb.4.0;Data Source=";  
strConnection+=Server.MapPath("*.mdb"); /*就是数据库的名字  
OleDbConnection objConnection=new OleDbConnection(strConnection);  
objConnection.Open();  
dim objConnection as OleDbConnection  
objConnection=new OleDbConnection("Provider=Microsoft.Jet.OleDb.4.0;Data Source="+Server.MapPath("*.mdb"))  
objConnection.Open()
```

下面再看一下SQL SERVER的连接数据库并打开；

```
string strConnection="server=数据库连接;uid=用户名;pwd=密码;database=数据库名字";  
SqlConnection objConnection=new SqlConnection(strConnection);  
objConnection.Open();  
dim objConnection as SqlConnection  
objConnection=new SqlConnection("server=数据库连接;uid=用户名;pwd=密码;database=数据库名字")  
objConnection.Open()
```

实际上，在大多数地方SQL SERVER和ACCESS的区别除了连接语句，其他定义语句也就是SQL××和OLEDB××的区别另外，如果是ACCESS数据库的话在ASPX文件的开头需要包括下列语句：

```
<%@Import Namespace="System.Data"%>  
<%@Import Namespace="System.Data.OleDb"%>
```

如果是SQL SERVER则需要包括以下语句：

```
<%@Import Namespace="System.Data"%>  
<%@Import Namespace="System.Data.SqlClient"%>
```

今天就说到这里，明天开始讲数据库的读取。

十天学会ASP.NET第六天

学习目的：学会读取数据库

有的网友说，不会C#，在ASP.NET里面还是选择VB.NET，这个我不反对，但是我觉得既然是学习新的东西，就不要在意有多少是新的。实际上C#就一般的操作来说，只是区分大小写和多了个分号这两点与VB.NET不同其他都差不多啊。在ASP里面我们读取数据就直接用RECORDSET，然后在HTML代码里面穿插<%= %>就可以显示了，但是ASP.NET讲究的是代码分离，如果还是这样显示数据就不能代码分离了。所以我们有两种方法：如果是读取一条记录的数据或者不多的数据，我们用DATAREADER采集数据，然后赋值给LABEL控件的Text属性即可；如果是读取大量数据我们就采用DATAGRID。

今天我们就来说一下DATAREADER：

```
C# string strConnection="Provider=Microsoft.Jet.OleDb.4.0;Data Source=";  
strConnection+=Server.MapPath(strDb);  
OleDbConnection objConnection=new OleDbConnection(strConnection);  
OleDbCommand objCommand = new OleDbCommand("这里是SQL语句", objConnection);  
objConnection.Open();  
OleDbDataReader objDataReader=objCommand.ExecuteReader();  
if(objDataReader.Read())  
{  
    oicq.Text=Convert.ToString(objDataReader["useroicq"]);  
    homesite.Text=Convert.ToString(objDataReader["usersite"]);  
    face.SelectedItem.Text=Convert.ToString(objDataReader["userface"]);  
}
```

大家可以看到我们首先是连接数据库然后打开，对于select的命令，我们申明一个OleDbCommand来执行之，然后再申明一个OleDbDataReader，来读取数据，用的是ExecuteReader()，objDataReader.Read() 就开始读取了，在输出的时候我们要注意Text属性接受的只能是字符串，所以我们要把读出的数据都转化为字符串才行。

转换变量类型函数：

转换为字符串：Convert.ToString()

转换为数字：Convert.ToInt64()，Convert.ToInt32()，Convert.ToInt16() 是按照数字位数由长到短

转换为日期：Convert.ToDateTime()

```
VB dim objConnection as OleDbConnection  
dim objCommand as OleDbCommand  
dim objDataReader as OleDbDataReader  
objConnection=new OleDbConnection(Provider=Microsoft.Jet.OleDb.4.0;Data Source="+Server.MapPath(strDb))  
objCommand=new OleDbCommand("这里是SQL语句", objConnection)  
objConnection.Open()  
objDataReader=objCommand.ExecuteReader()  
if objDataReader.Read()  
    oicq.Text=Convert.ToString(objDataReader["useroicq"])  
    homesite.Text=Convert.ToString(objDataReader["usersite"])  
    face.SelectedItem.Text=Convert.ToString(objDataReader["userface"])  
end if
```


其实大家比较一下C#和VB的语法，会觉得用C#似乎更加简单，教程里面我侧重C#，对于VB的代码我就不讲了。

下面说一下如果代码是读取SQL数据库，我们这样来转变

1、把代码开始的

```
<%@Import Namespace="System.Data"%>
```

```
<%@Import Namespace="System.Data.OleDb"%>
```

修改为

```
<%@Import Namespace="System.Data"%>
```

```
<%@Import Namespace="System.Data.SqlClient"%>
```

2、把代码里面所有声明的对象OleDb××变为Sql××就这么简单

你甚至可以用替换所有来解决，所以今后我讲不再列出Sql Server的代码了。

今天就说到这里，明天开始说数据的添加、删除、修改。

十天学会ASP.NET第七天

学习目的：学会添加删除修改记录

以前我们在ASP中不管是添加还是删除记录或者用SQL语句加上EXECUTE()或者就是直接用RECORDSET的NEW、DELETE、UPDATE等方法来添加删除修改记录。在ASP.NET里面我们采用SQL语句加上COMMAND对象来执行SQL语句的方法来添加删除修改记录。这比上次说的读取记录还简单。实际上只有一点区别，把ExecuteReader()修改为ExecuteNonQuery()即可，因为我们不需要返回值啊。

C#

```
string strConnection="Provider=Microsoft.Jet.OleDb.4.0;Data Source=";  
strConnection+=Server.MapPath(strDb);  
OleDbConnection objConnection=new OleDbConnection(strConnection);  
OleDbCommand objCommand = new OleDbCommand("这里是SQL语句", objConnection);  
objConnection.Open();  
OleDbDataReader objDataReader=objCommand.ExecuteNonQuery();
```

VB

```
dim objConnection as OleDbConnection  
dim objCommand as OleDbCommand  
objConnection=new OleDbConnection(Provider=Microsoft.Jet.OleDb.4.0;Data Source="+Server.MapPath(strDb))  
objCommand=new OleDbCommand("这里是SQL语句", objConnection)  
objConnection.Open()  
objDataReader=objCommand.ExecuteNonQuery()
```

下面再举例子说明一下SQL语句（ACCESS）：

添加记录：insert into 表名 (字段1， 字段2) values (字符串1， 字符串2) where 条件

删除记录：delete from 表名 where 条件

修改记录：update 表名 set 字段1=××， 字段2=×× where 条件

需要特别注意，如果是Sql Server删除记录是delete 表名，不需要再加from了。

今天就说到这里了，明天说ASP.NET里面最著名的DATAGRID。

十天学会ASP.NET第八天

学习目的：初步掌握DATAGRID的使用

以前在ASP中，我们显示大批量数据经常使用循环读取记录集的方式，在表格中插入代码来显示，循环的是<tr>，如果是分页还需要自己制作，如果排序更加复杂，在ASP.NET中就好了，一切工作都可以让DATAGRID来做。

首先看一下DATAGRID的样式方面的属性

BackImageUrl="" 背景图片

CellSpacing="" 单元格间距

CellPadding="" 单元格填充

cssClass="" 使用的CSS样式

DATAGRID可以自动用表中的字段名放在显示的记录的头部来表示各个单元格所代表的意义，用ShowHeader="true/false"来控制是不是显示，在大多数情况下我们是不需要这个功能的，因为我们数据库中的字段名大多是英文的，而我们想在页面输出的大多是中文名字。

下面看一下显示数据库内所有记录，就几行代码：

```
<script runat="server" language="c#">
void Page_Load()
{
string strConnection="Provider=Microsoft.Jet.OleDb.4.0;Data Source=";
strConnection+=Server.MapPath("guestbook.mdb");
OleDbConnection objConnection=new OleDbConnection(strConnection);
OleDbCommand objCommand1 = new OleDbCommand("select * from guestbook",objConnection);
objConnection.Open();
dgrdMain.DataSource=objCommand1.ExecuteReader();//dgrdMain就是下面的DATAGRID的ID
dgrdMain.DataBind();
objConnection.Close();
}
</script>
<html>
<body>
<asp:DataGrid
id="dgrdMain"
cellpadding="1"
showheader="true"
borderwidth="0"
runat="server"
/>
</body>
</html>
```

至于VB的版本，这次大家自己来试试吧：)

假设数据库是三个字段：id,aa,bb

显示的样子就如下：

id	aa	bb
1	werwe	rewrwe
2	werwe	rewrwe

我们一定觉得这种显示不满意，我们有两种显示的方式（需要包括在<columns> </columns>中间）

一、默认的列，我们可以选择不输出所有的字段还可以安排顺序：

```
<asp:BoundColumn DataField="想要显示的字段名">
```

比如说我们想按照bb,aa来输出这个表，我们这么写

```
<asp:DataGrid  
id="dgrdMain"  
cellpadding="1"  
showheader="true"  
autogeneratecolumns="false"  
borderwidth="0"  
runat="server">  
<columns>  
<asp:boundcolumn datafield="bb"/>  
<asp:boundcolumn datafield="aa"/>  
</columns>  
</asp:datagrid>
```

注意使用了 autogeneratecolumns="false" 以后DATAGRID就不会自动输出所有字段了。

二、以模板列，我们可以定制每一个单元格的样式：

```
<asp:TemplateColumn>
```

```
<itemTemplate>
```

中间是一个表格，想怎么样就怎么样

```
</itemTemplate>
```

```
</asp:DataGrid>
```

或许你们要说怎么在表格中输出代码那，可以使用<%# DataBinder.Eval(Container.DataItem,"字段名").ToString()%>

下面看一个例子，我们所要做的就是将aa,bb两个字段放在一个单元格内显示，就好像下面一样：

1	werwe rewrwe
2	werwe rewrwe

我们这样写代码：

```
<asp:DataGrid  
id="dgrdMain"  
cellpadding="1"  
showheader="false"  
autogeneratecolumns="false"  
borderwidth="0"  
runat="server">  
<columns>  
<asp:boundcolumn datafield="id"/> 首先使用默认的显示方式显示ID  
<asp:TemplateColumn> 再使用模板来显示后面一列（由AA,BB并列组成）  
<itemTemplate>  
<table border="0" cellspacing="0" cellpadding="0" width="100%">  
<tr>  
<td><%# Container.DataItem("aa")%></td>  
</tr>  
<tr>  
<td><%# Container.DataItem("bb")%></td>  
</tr>  
</table>  
</itemTemplate>  
</asp:DataGrid>  
</columns>  
</asp:datagrid>
```

今天就说到这里，明天我们来看一下DATAGRID的分页显示等功能

十天学会ASP.NET第九天

学习目的：学会用DATAGRID分页显示

虽然说DATAGRID的分页显示效率不怎么高，但是不管怎么说，它是最方便的分页显示，对于做一个留言簿还是相当简单实用的。对于启用分页功能还有一个重要要求，只能在DATADRID控件的数据源实现了ICOLLECTION接口的情况下才能启用分页功能，而DATAREADER没有这个接口，所以必须使用DATATABLE来代替。

```
C# <script runat="server" language="c#">
void Page_Load()
{
    string strConnection="Provider=Microsoft.Jet.OleDb.4.0;Data Source=";
    strConnection+=Server.MapPath("guestbook.mdb");
    OleDbConnection objConnection=new OleDbConnection(strConnection);
    OleDbDataAdapter objDataAdapter=new OleDbDataAdapter("select * from guestbook",objConnection);
    DataSet objDataSet=new DataSet();
    objDataAdapter.Fill(objDataSet);
    dgrdMain.DataSource=objDataSet;
    dgrdMain.DataBind();
}
void dgrdMain_PageIndexChanged(Object sender, DataGridPageChangedEventArgs e)
{
    dgrdMain.CurrentPageIndex = e.NewPageIndex;
    DataBind();
}
</script>
<html>
<body>
<asp:DataGrid
id="dgrdMain"
cellpadding="1"
showheader="true"
borderwidth="0"
allowpaging="true"  启用分页功能
pagesize="3"  每一页显示三条记录
onpageindexchanged="dgrdMain_pageindexchanged "  调用第×页的函数为dgrdMain_PageIndexChanged()
runat="server"
/>
</body>
</html>
```

```
VB <script runat="server" language="vb">
sub Page_Load
dim objConnection as OleDbConnection
dim objDataAdapter as OleDbDataAdapter
dim objDataSet as DataSet
objConnection=new OleDbConnection("Provider=Microsoft.Jet.OleDb.4.0;Data Source="+Server.
MapPath("guestbook.mdb"))
objDataAdapter=new OleDbDataAdapter("select * from guestbook",objConnection)
objDataSet=new DataSet
objDataAdapter.Fill(objDataSet)
```

```
dgrdMain.DataSource=objDataSet
dgrdMain.DataBind()
end sub
sub dgrdMain_PageIndexChanged(sender as object,e as datagridpagechangedeventargs)
dgrdMain.CurrentPageIndex = e.NewPageIndex
DataBind
end sub
</script>
<html>
<body>
<asp:DataGrid
id="dgrdMain"
cellpadding="1"
showheader="true"
borderwidth="0"
allowpaging="true" 启用分页功能
pagesize="3" 每一页显示三条记录
onpageindexchanged="dgrdMain_pageindexchanged " 调用第x页的函数为dgrdMain_PageIndexChanged()
runat="server"
/>
</body>
</html>
```

这样就是最简单的分页，我们可以用参数`pagestyle-mode="nextprev/numericpages"`来设定是显示上一页、下一页方式还是直接选择页面的数字方式来实现页面跳转。如果是前者我们可以用`pagestyle-nextpagetext`来设定包含下一页链接的文本，用`pagestyle-prevpagetext`来设定包含上一页链接的文本；如果是后者我们可以用`PagerStyle-PageButtonCount`来设定在使用省略号之前显示的页面编号数量（默认是10）

今天就说到这里，明天说一下SESSION和COOKIE。

十天学会ASP.NET第十天

学习目的：学会SESSION和COOKIE的使用

在各个网页间的变量传递和一些记录用户的登陆信息要用到SESSION和COOKIE。在ASP.NET中使用COOKIE比ASP中稍微麻烦一点，因为我们要申明变量。

首先看一下SESSION的使用，基本和ASP差不多

写入一个SESSION：

```
Session["username"] = "aa";
```

```
Session("username") = "aa"
```

读取一个SESSION：

```
string username = Session["username"];
```

```
dim username = Session("username")
```

再来看一下COOKIE的写入：

```
DateTime dt = DateTime.Now; //需要<%@Import Namespace="System"%>，得到当前时间
```

```
HttpCookie mycookie = new HttpCookie("logname"); //申明新的COOKIE变量
```

```
mycookie.Value = "aa"; //赋值
```

```
mycookie.Expires = Convert.ToDateTime(dt + TimeSpan.FromDays(1)); //设定过期时间为1天
```

```
Response.Cookies.Add(mycookie1); //写入COOKIE
```

```
dim dt as DateTime
```

```
dt = DateTime.Now
```

```
dim mycookie as HttpCookie
```

```
mycookie = new HttpCookie("logname")
```

```
mycookie.Value = "aa"
```

```
mycookie.Expires = Convert.ToDateTime(dt + TimeSpan.FromDays(1))
```

```
Response.Cookies.Add(mycookie1)
```

看一下COOKIE的读取：

```
HttpCookie mycookie = Request.Cookies["username"];
```

```
string username = mycookie.Value;
```

```
dim mycookie as HttpCookie
```

```
mycookie = Request.Cookies["username"]
```

```
dim string = mycookie.Value
```

到此《十天学会ASP.NET》结束了，时间仓促，还请大家原谅写的不全，实际上我只是带大家入门，学会还要靠自己啊。