

PAGINE ASP.NET

Nelle pagine seguenti impareremo ad implementare le **pagine ASP**, cioè quelle pagine che ci consentono di creare e sviluppare software in rete eseguibile lato server e che riconosciamo dall'estensione **.aspx**.

Codificare una pagina ASP

Per i nostri lavori utilizziamo l'ambiente Expression web. In alternativa si può usare Web developer express, Visual Studio, o un semplice editor come block notes (in questo caso dovremo però preoccuparci di installare il web server).

Una pagina ASP.NET inizia con la **direttiva Page** per la dichiarazione che identifica il linguaggio di implementazione scelto

```
<%@ Page language="VB" %>
```

oppure

```
<%@ Page language="C#" %>
```

Il VB è il linguaggio di default, quindi nel nostro caso la direttiva potrebbe essere omessa ed ha il solo scopo di una più chiara documentazione.

Nelle pagine ASP.NET il codice VB viene separato dai tag HTML e viene interamente implementato all'inizio della pagina, prima di HTML.

E' compreso tra il tag **<script runat="server">** con l'attributo obbligatorio **runat="server"** e la chiusura **</script>**.

Esso va interamente gestito in questa sezione, all'interno delle routines che verranno poi associate agli eventi dei controlli web, incluse tutte le istruzioni da eseguire al caricamento della pagina che vengono associate alla Sub di default Page_Load.

I Controlli

Le pagine ASP.NET utilizzano diversi gruppi di controlli, tra i quali:

HTML server controls

Web server controls

HTML server controls

Gli **HTML Server controls** sono delle classi che rappresentano elementi HTML in ambiente .NET. Sono definiti nel namespace System.Web.UI.HtmlControls, implicitamente utilizzato dal file aspx in esecuzione, e le classi che li descrivono mantengono dichiarazioni e attributi tipici dei classici tag HTML. Essi consentono dunque ai programmatori abituati alla programmazione HTML di mantenere le proprie sintassi in modo semplice e compatibile con il passato, quindi molto utili nelle operazioni di migrazione.

E' possibile utilizzare un controllo server HTML aggiungendo l'attributo **runat=server** ai più comuni tag, dotandoli di **proprietà ed eventi**, gestendoli via codice e simulando un ambiente di programmazione tradizionale.

Di seguito l'elenco

HTML Server Control	Descrizione
HtmlAnchor	elemento HTML <a>
HtmlButton	elemento HTML <button>
HtmlForm	elemento HTML <form>
HtmlGeneric	altri ementi HTML non specifici come <body>, <div>, , etc.
HtmlImage	elemento HTML <image>
HtmlInputButton	elementi HTML <input type="button">, <input type="submit"> , <input type="reset">
HtmlInputCheckBox	elemento HTML <input type="checkbox">
HtmlInputFile	elemento HTML <input type="file">
HtmlInputHidde	elemento HTML <input type="hidden">
HtmlInputimage	elemento HTML <input type="image">
HtmlInputRadioButton	elemento HTML <input type="radio">
HtmlInputText	elementi HTML <input type="text"> e <input type="password">
HtmlSelect	elemento HTML <select>
HtmlTable	elemento HTML <table>
HtmlTableCell	elementi HTML <td> e <th>
HtmlTableRow	elemento HTML <tr>
HtmlTextArea	elemento HTML <textarea>

Ecco di seguito alcuni semplici esempi

Due pagine di esempio dell'uso del tag **<a href >** per linkare a una pagina2.aspx. Nel primo esempio usiamo il tag in modo del tutto tradizionale, mentre nel secondo ne gestiamo le proprietà nella Sub Page_Load

htmlAnchoresempio1.aspx

```
<%@ Page Language="VB" %>
<script language="vb" runat="server">
</script>
<html>
<head>
<title> esempio </title>
</head>
<body>
<form runat="server">
<a id="link" href="pagina2.aspx" runat="server">
questo è un primo esempio di html Anchor
</a>
</form>
</body>
</html>
```

htmlAnchoresempio2.aspx

```
<script language="vb" runat="server">
  Sub Page_Load(sender as Object, e as EventArgs)
    link.href = "pagina2.aspx"
    link.title="Questo è un secondo esempio di html server controls"
  End Sub
</script>
<html>
<head>
<title>esempio</title>
</head>
<body>
<form runat="server">
  <a id="link" runat="server">
    esempio di html Anchor
  </a>
</form>
</body>
</html>
```

Esempio di uso del **controllo text e submit** con gestione di proprietà ed eventi :l'evento onserverclick sul control submit genera l'esecuzione della Sub scritta che assegna un valore al control text.

```
<%@ Page Language="VB" %>
<script runat="server">
Sub scritta( ByVal sender As Object, ByVal e As EventArgs)
nome.value="hello world"
End Sub
</script>
<html>
<head>
<title>html controls</title>
</head>
<body>
<form id="form1" runat="server">
<input id="nome" type="text" runat="server" />
<br /> <br /> <br />
<input id="submit1" type="submit" value="visualizza la scritta"
onserverclick="scritta" runat="server" />
</form>
</body>
</html>
```

Web server controls

I **Web Server controls** sono oggetti che forniscono astrazione di alto livello e una funzionalità più ampia rispetto agli HTMLControls, potendo generare insiemi più numerosi di eventi che necessiterebbero decine di righe di codice HTML. Possono dunque rappresentare elementi HTML esistenti o rappresentare elementi più complessi.

I Web controls si trovano nel namespace System.Web.UI.WebControls ed ereditano dalla classe WebControl , contenuta nello stesso namespace.

La sintassi per creare un Web server control è

```
<asp:control_name id="some_id" runat="server" />
```

Di seguito l'elenco

Web Server Control	Descrizione
AdRotator	sequenza di immagini
Button	Command button
Calendar	calendario
CalendarDay	giorno in un calendar control
CheckBox	check box
CheckBoxList	gruppo di check per multi-selezione
DataGrid	griglia di campi associata a un data source
DataList	elementi di un data source con uso ditemplates
DropDownList	list drop-down
Hyperlink	hyperlink
Image	immagine
ImageButton	immagine cliccabile
Label	output statico programmabile e con stile applicabile al contenuto
LinkButton	hyperlink button
ListBox	list drop-down con selezione singola o multipla
ListItem	elemento di un list
Literal	output statico programmabile
Panel	container per altri controlli
Placeholder	riserva spazio per controlli aggiunti da codice
RadioButton	radio button
RadioButtonList	gruppo di radio buttons
BulletedList	list in bullet format
Repeater	elenco ripetuto di elementi legati al controllo
Style	imposta lo style di controlli
Table	table
TableCell	table cell
TableRow	table row
TextBox	text box
Xml	file XML o risultati di trasformazione XSL

Ecco di seguito alcuni semplici esempi :

Pagina per eseguire la classica somma di due numeri

```
<%@ Page Language="VB" %>
<script runat="server">
  Sub Page_Load(ByVal sender as Object, e as EventArgs)
    lbltitolo.text = "somma di due numeri"
  End Sub
Sub Calcola (ByVal sender as Object, e as EventArgs)
  Dim somma as Double
  somma=val(num1.text)+val(num2.text)
  lblrisultato.text=" la somma è " & somma
End Sub
Sub Reset (ByVal sender as Object, e as EventArgs)
  num1.text=""
  num2.text=String.Empty
  lblrisultato.text=""
end Sub
</script>

<html>
<head>
<title>somma di due numeri</title>
</head>
<body>
<form runat="server">
<p>
<asp:label id="lbltitolo" runat="server"
  forecolor="red"
  font-bold="True" />
</p>
<p>
  primo numero :<asp:TextBox id="num1" runat="server" />
  secondo numero :<asp:TextBox id="num2" runat="server" />
  <asp:Button id="bt1" OnClick="Calcola" text="esegui" runat="server" />
  <asp:Button id="bt2" OnClick="Reset" text="annulla" runat="server" />
</p>
<p> <asp:label id="lblrisultato" runat="server" /> </p>
</form>
</body>
</html>
```

In questa pagina abbiamo utilizzato diversi controlli: label (lbltitolo ed lblrisultato), textbox (num1 e num2) e Button (bt1 e bt2), dichiarati secondo la sintassi

```
<asp:control_name id="some_id" runat="server" />
```

ne abbiamo impostato alcune proprietà come forecolor, font-bold o text.

Abbiamo definito tre sottoprogrammi:

il **Page_Load**, routine di default che gestisce l'evento del caricamento della pagina, cioè le azioni da compiere quando il browser apre la pagina, e abbiamo assegnato alla lbltitolo la proprietà text
il sottoprogramma Calcola che abbiamo associato all'evento OnClick del button bt1 per acquisire i

dati nei due TextBox, eseguire e visualizzare la somma nella label lblrisultato
ilsottoprogramma Reset associato all'evento OnClick del button bt2 per ripulire i text e la label
assegnando "" o String.Empty.

Questo secondo script visualizza data e ora di sistema

```
<%@ Page Language="VB" %>
<script runat="server">
Sub Page_Load(ByVal sender as Object, e as EventArgs)
Dim dataora as DateTime=Now()
lbldata.text=dataora.toString()
End Sub
</script>
<html>
<head>
<title>Data e ora</title>
</head>
<body>
<form runat="server">
<asp:label id="lbldata" runat="server" />
</form>
</body>
</html>
```

Nella routine Page-Load abbiamo assegnato ad una variabile di tipo DateTime la funzione Now() che restituisce la data e l'ora corrente. Abbiamo associato la variabile alla proprietà text della label lbldata, convertendola in stringa con il metodo toString().

ACCESSO AI DATABASE

Negli esempi successivi impareremo a connetterci e a gestire un database su server con l'uso della tecnologia ADO.NET.

Per approfondimenti sulla struttura di ADO.NET rimandiamo alle lezioni sulla gestione dei dati in windows form. E' evidente tuttavia che la situazione inerente alle applicazioni web è ben diversa dalle applicazioni desktop, e può generare una serie di **problematiche, quali problemi di stato, di scala o di compatibilità, legati alla natura disconnessa del protocollo HTTP, alla simultaneità degli accessi ad opera di più utenti e agli eventuali conflitti**. Quindi è necessario prestare molta attenzione, alla scala della nostra applicazione, all'impiego della memoria del server, alla connessione al database, soprattutto in caso di applicazioni con modifiche di notevole entità.

Gli oggetti principali per costruire siti web con gestione di database sono
l'oggetto **Connection** per stabilire la connessione al database
l'oggetto **Command** per eseguire i comandi di gestione del database
l'oggetto **DataReader** per avere disponibili i risultati delle interrogazioni al database

Prima di procedere creiamo sotto la directory del localhost una ulteriore cartella che conterrà tutte le pagine per la gestione del database e un database in Access chiamato db con una semplice tabella di nome iscritti i cui campi di tipo testo sono nick, pw, email.

Per rendere il nostro lavoro sempre coerente abituiamoci a linkare tutte le pagine con una pagina di avvio, che amplieremo nel corso del nostro percorso.

Impostiamo quindi un file chiamato default.aspx (equivalente del più noto index) simile a questo:

```
<html>
<head>
<title>Gestione di un database</title>
</head>
<body>
<p><a id="link" href="leggi.aspx" runat="server">pagina di stampa </a></p>
<p><a id="link2" href="inserisci.aspx" runat="server">pagina di inserimento </a></p>
</body>
</html>
```

Effettuare la connessione al database

In ADO.NET l'oggetto **Connection** rappresenta una **connessione all'origine dati**. Per prima cosa dobbiamo importare i riferimenti relativi ai namespace per il managed provider che vogliamo usare, nel nostro caso lo standard ODBC, anche in vista della dismissione di OleDb annunciata da Microsoft. Inseriamo dunque le direttive

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

Successivamente possiamo dichiarare l'oggetto Connection, impostare la stringa di connessione al database e infine aprire la connessione

```
<script runat="server">
Sub Page_Load(Sender as Object, E as EventArgs)
Dim con As New Odbc.OdbcConnection()
con.ConnectionString = "Driver={Microsoft Access Driver (*.mdb)};DBQ="+Server.MapPath("db.mdb")+";"
con.Open()
End Sub
</script>
```

questo codice va ripetuto per ogni pagina del sito che interagisca con il DataBase

Stampa del contenuto di una tabella del database

Questa pagina denominata leggi.aspx consente la stampa dei dati della tabella iscritti. I passi da codificare sono

apertura della connessione con il database db.mdb

creazione dell'apposita query sql

associazione di un oggetto command alla query e alla connessione

esecuzione del comando con associazione al datareader

ciclo while di scorrimento del datareader e visualizzazione dei record

```

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

<script runat="server">
Sub Page_Load(ByVal Sender as Object, E as EventArgs)
Dim con As New Odbc.OdbcConnection()
con.ConnectionString = "Driver={Microsoft Access Driver (*.mdb)};DBQ="+Server.MapPath("db.mdb")+";"
con.Open()
msg.Text = "elenco iscritti"
dim query as string
query="select nick, pw from iscritti;"
Dim cmd as New odbc.odbcCommand(query,con)
Dim myDataReader as OdbcDataReader
myDataReader = cmd.ExecuteReader()
While myDataReader.Read()
lst.items.add( mydatareader("nick") + " " +myDataReader("pw"))
end while
myDataReader.Close()

End Sub
</script>
<html>
<head>
<title>Stampa della Tabella</title>
</head>
<body>
<form runat=server>
<h1>
<asp:Label id="msg" runat="server"></asp:Label>
</h1>
<asp:listbox id="lst" runat="server" Height="118px" Width="132px"></asp:listbox>
<a id="link" href="default.aspx" runat="server" >home</a>
</form>
</body>
</html>

```

Inserimento di un record nella tabella del database

Questa pagina denominata inserisci.aspx consente l'inserimento di un nuovo record nella tabella.

I passi da codificare sono

apertura della connessione con il database db.mdb

creazione dell'apposito comando sql di insert con acquisizione dei dati da client

associazione di un oggetto command alla query e alla connessione

esecuzione del comando con l'istruzione executenonquery

```

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

<script runat="server">

```



```
Sub Page_Load(ByVal Sender as Object, E as EventArgs)
```

```
    msg.Text = "inserimento nuovo record"
```

```
End Sub
```

```
Sub inserisci (Sender as Object, E as EventArgs)
```

```
Dim con As New Odbc.OdbcConnection()
```

```
con.ConnectionString = "Driver={Microsoft Access Driver (*.mdb)};DBQ="+Server.MapPath("db.mdb")+";"
```

```
con.Open()
```

```
    dim query as string
```

```
    query="insert into iscritti (nick, pw, email) values ('"+txtnick.text+"','"+txtpw.text+"','"+txtemail.text+"');"
```

```
    Dim cmd as New odbc.odbcCommand(query,con)
```

```
    cmd.ExecuteNonQuery
```

```
End Sub
```

```
</script>
```

```
<html>
```

```
<head>
```

```
    <title>inserimento record</title>
```

```
</head>
```

```
<body>
```

```
<form runat=server>
```

```
<h1>
```

```
<asp:Label id="msg" runat="server" />
```

```
</h1>
```

```
nick <asp:TextBox id="txtnick" runat="server" />
```

```
pw <asp:TextBox id="txtpw" runat="server" />
```

```
email <asp:TextBox id="txtemail" runat="server" />
```

```
<br /> <br />
```

```
<asp:Button id="bt1" onclick="inserisci" Text="inserisci record" runat="server" Height="41px" />
```

```
    <br /><br /><br />
```

```
<a id="link" href="default.aspx" runat="server" >home</a>
```

```
</form>
```

```
</body>
```

```
</html>
```