

TIPO: ASPAGGIORNARE UN DATABASE

```
<% 'on error resume next
dim conaa
set conaa = server.createobject("adodb.connection")
conaa.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" &
server.mappath("/cartella/dati.mdb")
set rsaa = server.createobject("adodb.recordset")
Dim SQL
SQL = "UPDATE blog SET commenti=" & punti & " WHERE id=" & inmes & ""
rsaa.Open SQL, conaa, 3, 3
'response.write(punti)
set rsaa= nothing
set conaa = nothing %>
```

TIPO: ASPAGGIORNARE UN DATABASE SENZA RECODSET

```
dim sql
sql="UPDATE immobili SET
immobile=" & immobile & ",zona=" & zona & ",indirizzo=" & indirizzo & ",tipo=" & tipo & ",motivo=" & motivo &
",mq=" & mq & ",prezzo=" & prezzo & ",descrizione=" & descrizione & " WHERE codice=" & codice & ""

set conn = server.createobject("adodb.connection")
conn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" &
server.mappath("/agenzia/dati/immobili.mdb")
'set rs = server.createobject("adodb.recordset")
'rs.open sql, conn ,3 ,3
conn.execute(sql)
'rs.update
'rs.movelast
'rs.close
'set rs= nothing
set conn = nothing
```

TIPO: ASPALTERNARE I COLORI NELLE TABELLE ASP

```
<!--Apriamo il Tag della tabella-->
<table>
<%
strsql = "select * from Nome_Tabella"
set rs = con.execute(strsql)
conta = 0
while not rs.eof
bgcolor = "white"
if conta mod 2 <> 0 then bgcolor = "lightblue"
%>
<!--Adesso i tag per le celle della tabella-->
<tr>
<td bgcolor="<%=bgcolor%>"><%=rs("nome_campo")%></td>
</tr>
<%
conta = conta + 1
rs.movenext
wend
rs.close
set rs = nothing
%>
<!--Chiudiamo il tag della tabella-->
</table>
```

TIPO: ASP

ANDARE A CAPO IN UNA STRINGA CON ASP

```
<% dim testo
testo=request.form("testo1")
dim format
format=replace(testo, chr(10), "<br>")
response.write(format)
%>
```

TIPO: ASP

APRIRE UN DATABASE IN BASE AD UN DATO E REDIRECT SE NON C'È RISPOSTA

```
<%
Set Conn = Server.CreateObject("ADODB.Connection")
conn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" &
server.mappath("/utenti/messaggi.mdb")
Dim rs
Set rs = Server.CreateObject("ADODB.Recordset")
rs.Open "select * from utenti where userepass ='"&userepass&'"", conn
if rs.eof then Response.Redirect("http://www.filoweb.it/utenti/inserisci.htm")
Do while NOT RS.EOF %>
<%
RS.MoveNext
Loop
RS.Close
Set RS = Nothing
conn.Close
Set conn=Nothing
%>
```

TIPO: ASP

CANCELLARE I DATI DA UN DATABASE

```
<%
on error resume next
dim conn
set conn = server.createobject("adodb.connection")
conn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" &
server.mappath("/test/preventivi.mdb")
set rs = server.createobject("adodb.recordset")
rs.open "DELETE * FROM prodotti WHERE codice='"&risult&'"",conn
rs.close
set rs= nothing
set conn = nothing

%>
```

TIPO: ASP

CANCELLARE I RECORD SELEZIONATI IN UN CHECKBOX

```
Parte che invia
<form id="form1" name="form1" method="post" action="cancella.asp">
<%Dim Conn
Set Conn = Server.CreateObject("ADODB.Connection")
conn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" &
server.mappath("/sitoasp/sondaggio.mdb")
Dim rs
Set rs = Server.CreateObject("ADODB.Recordset")
rs.Open "select * from domande ", conn %>
<table width="400" border="1" cellpadding="0" cellspacing="0">
<tr><td width="63" height="27">cancella</td>
<td width="189" bgcolor="#CCCCCC"><div align="left" class="style3">Domanda</div></td><td
width="140" bgcolor="#CCCCCC"><div align="center" class="style3">Punteggio</div></td></tr>
<% Do while NOT RS.EOF %>
<%Dim domanda
```

```

domanda=rs("domanda")%>
  <% dim id
id=rs("id")%>
<%response.write("<tr><td><input name=cancella type=checkbox value="&id&"
/></td><td>"&domanda&" </td><td> "&punteggio&"</td></tr>")%></br>
<% RS.MoveNext
Loop %> </input> </table>
<% RS.Close
Set RS = Nothing
conn.Close
Set conn=Nothing %>
  <label> <input type="submit" name="Submit" value="Submit" /> </label>
  </form>
</blockquote>
</blockquote>
</body>
</html>

```

Parte che riceve

```

<%
on error resume next
  dim conb
  set conb = server.createobject("adodb.connection")
conb.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" &
server.mappath("/sitoasp/sondaggio.mdb")
set rsb = server.createobject("adodb.recordset")
Dim objColor
for each objColor in Request.Form("cancella")
Response.Write("Ho cancellato il messaggio numero: " & objColor & "<br>")
rsb.open "DELETE * FROM domande WHERE id="&objColor&"",conb
next
rsb.close
set rsb= nothing
set conb = nothing
%>

```

TIPO: ASP

CANCELLARE UN FILE DAL SERVER

```

<%
Dim objFSO
Set objFSO = Server.CreateObject("Scripting.FileSystemObject")
'Il file viene cancellato
objFSO.DeleteFile("c:\freeasp.txt")

Set objFSO = Nothing
%>

```

TIPO: ASP

CANCELLARE UNA CARTELLA

```

<%
'dichiarazione variabili
Dim FSO
dim nomecartella
nomecartella="nome_cartella"
SET FSO = Server.CreateObject("Scripting.FileSystemObject")
FSO.DeleteFolder(Server.MapPath(nomecartella))
SET myFSO = NOTHING
%>

```

TIPO: ASP

CARICARE UN'IMMAGINE RANDOM DA UNA DIRECTORY

```

<% Const IMGS_DIR = "/gallery/GALLERIA/thumbnails/"

```

```

Dim objFSO, objFolderObject, objFileCollection, objFile
Dim intFileNumberToUse, intFileLooper
Dim objImageFileToUse
Dim strImageSrcText
Set objFSO = Server.CreateObject("Scripting.FileSystemObject")
Set objFolderObject = objFSO.GetFolder(Server.MapPath(IMGS_DIR))
Set objFSO = Nothing
Set objFileCollection = objFolderObject.Files
Set objFolderObject = Nothing
Randomize()
intFileNumberToUse = Int(objFileCollection.Count * Rnd) + 1
intFileLooper = 1
For Each objFile in objFileCollection
    If intFileLooper = intFileNumberToUse Then
        Set objImageFileToUse = objFile
        Exit For
    End If
    intFileLooper = intFileLooper + 1
Next
Set objFileCollection = Nothing
strImageSrcText = IMGS_DIR & objImageFileToUse.Name
Set objImageFileToUse = Nothing
%>

```

TIPO: ASP

COMANDI E FUNZIONI

InStr: Restituisce la posizione della prima occorrenza di una stringa dentro un'altra. La ricerca inizia con il primo carattere della stringa.

InStrRev :Restituisce la posizione della prima occorrenza di una stringa dentro un'altra. La ricerca inizia con l'ultimo carattere della stringa.

Lcase :Converte una stringa in minuscolo

Left : Restituisce un certo numero di caratteri a partire dal lato sinistro della stringa

Mid :Restituisce un certo numero di caratteri da una stringa

Right :Restituisce un certo numero di caratteri a partire dal lato destro della stringa

Len: Restituisce il numero di caratteri nella stringa.

Ltrim: Rimuove gli spazi nel lato sinistro della stringa

Rtrim: Rimuove gli spazi nel lato destro della stringa

Trim: Rimuove gli spazi da entrambe le estremità della stringa

Replace: Sostituisce una parte della stringa con un'altra.

Space: Restituisce una stringa formata da un determinato numero di spazi

StrComp: Confronta due stringhe e restituisce un valore che rappresenta il risultato del confronto

String: Restituisce una stringa che contiene uno specifico carattere ripetuto un certo numero di volte

StrReverse: Inverte la stringa

Ucase: Converte una stringa in maiuscolo

Month(date): Restituisce il mese corrente

Day(data): Restituisce il giorno corrente

Year(date): Restituisce l'anno corrente

Left(stringa, lunghezza): Restituisce i caratteri in una stringa

TIPO: ASP

CREARE UNA CARTELLA CON ASP

```
<%
```

```
Dim FSO, ind, var
```

```
ind = nome cartella principale
```

```
var = nome cartella da creare
```

```
SET FSO = Server.CreateObject("Scripting.FileSystemObject")
```

```
'L'indirizzo lo "creiamo" dinamicamente con l'utilizzo di 2 variabili,
```

```
' - ind : contiene la cartella in cui ci si trova
```

```
' - var : il nome della nuova cartella
```

```
FSO.CreateFolder(Server.MapPath(ind & "\" & var)
SET FSO = NOTHING
%>
```

TIPO: ASP

CREARE UN'ALBUM DI IMMAGINI CO I FILE DI IMMAGINE IN UNA CARTELLA

```
<%
Set fileCreate = CreateObject("Scripting.FileSystemObject")
Set fileGet = fileCreate.GetFolder(Server.MapPath("/immagine"))
Set files = fileGet.Files %>
```

```
<% For Each file in files
  response.Write("</p>")
Next
Set fileCreate=Nothing
Set fileGet=Nothing
Set files=Nothing
%>
```

TIPO: ASP

CREARE UN'ALBUM DI IMMAGINI CO I FILE DI IMMAGINE IN UNA CARTELLA CON TUNBAIL

```
<%
Set fileCreate = CreateObject("Scripting.FileSystemObject")
Set fileGet = fileCreate.GetFolder(Server.MapPath("/immagine"))
Set files = fileGet.Files %>
```

```
<% For Each file in files
  response.Write("<a href= " &file.name & "> </a></p>")
Next
Set fileCreate=Nothing
Set fileGet=Nothing
Set files=Nothing
%>
```

TIPO: ASP

DATA DA INGLESE A ITALIANO IN PAGINE ASP

```
<%
Dim original
original = SetLocale(1040)

'Escludo i secondi dall'ora
ora = Mid(Time(),1,5)
'Aggiungo alla data in formato lungo, l'ora estrapolata
datone = FormatDateTime(Date(), vbLongDate) & " " & ora
Response.Write("datone=" & datone)
%>
```

TIPO: ASP

DIVIDERE IL RISULTATO DI UNA QUERY IN PIÙ PAGINE

Supponiamo, per esempio, di avere un database Access contenente gli iscritti ad una mailing list, e di voler visualizzare tutti gli iscritti sul nostro browser; poniamo di voler visualizzare 5 iscritti in ogni pagina. Il codice del file, che chiameremo list.asp, è qui di seguito visualizzato:

```
<%
'Creiamo la connessione al database
Dim Conn
Set Conn = Server.CreateObject ("ADODB.Connection")
Conn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" & _
  server.MapPath("maillinglist.mdb")
```

```
'Rileviamo la pagina da visualizzare dalla stringa dell'URL
pag = TRIM(Request.QueryString("pag"))
If pag="" Then
    pag = 1
Else
    pag = CInt(pag)
End If
```

```
'Creiamo la stringa sql
Dim sql
sql = "SELECT * FROM Tabella_Iscritti"
Dim rs
Set rs = Server.CreateObject ("ADODB.Recordset")
'Settiamo il numero di record per pagina
rs.PageSize = 5
rs.Open sql, conn, 3, 3
rs.AbsolutePage = pag
```

```
'Ora stampiamo gli indirizzi su schermo
For i=1 to rs.PageSize
    Response.Write( rs("NOME")&"<br>" )
    rs.MoveNext
    If rs.EOF Then Exit For
Next
```

```
'Prepariamo i link per le altre pagine della lista
If ( pag > 1) Then
    Response.Write("<a href=""list.asp?pag=" &pag-1& "">Pagina precedente</a>")
End If
If ( pag < rs.PageCount ) Then
    Response.Write( "<a href=""list.asp?pag=" &pag+1& "">Pagina successiva</a>")
End If
%>
```

Nell'esempio abbiamo sfruttato alcune proprietà dei recordset:

- PageCount: ci permette di sapere il numero totale delle pagine;
- AbsolutePage: imposta o restituisce la pagina corrente;
- PageSize: consente di impostare il numero di record che si intende visualizzare in ogni pagina.

TIPO: ASP

ESEMPIO DI IF ED ELSEIF

```
<%
```

```
a = 0
b = 1
```

```
If a = b then
    ris ="a e b sono uguali"
Elseif a > b then
    ris ="a è maggiore di b"
Elseif a < b then
    ris ="a è minore di b"
End If
```

```
%>
```

TIPO: ASP

FILES IN UNA CARTELLA

```
<%
```

```
Set fileCreate = CreateObject("Scripting.FileSystemObject")
Set fileGet = fileCreate.GetFolder(Server.MapPath("/cartella"))
```

```

Set files = fileGet.Files
For Each file in files
response.write "<A HREF=' "
response.write file.name
response.write " '>"
response.write file.name
response.write "</A> " & file.size & " Kb "
Next
Set fileCreate=Nothing
Set fileGet=Nothing
Set files=Nothing
%>

```

TIPO: ASP

GENERARE UNA PASSWORD CASUALE (NUMERI E LETTERE)

```

<%
caratteri = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
Randomize ' inizializza il generatore di numeri casuale
Do Until len(password) = 10 ' qui imposti la lunghezza della password
carattere = Int((37 * Rnd) + 1) ' Genera un valore casuale compreso tra 0 e Z.
password = password & Mid(caratteri,carattere,1)
Loop
Response.Write ("La tua password è : <b>" & password & "</b>")
%>

```

TIPO: ASP

GENERARE UNA PASSWORD CASUALE (SOLO LETTERE)

```

<%
caratteri = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
Randomize ' inizializza il generatore di numeri casuale
Do Until len(password) = 10 ' qui imposti la lunghezza della password
lettera = Int((27 * Rnd) + 1) ' Genera un valore casuale compreso tra A e Z.
password = password & Mid(caratteri,lettera,1)
Loop
Response.Write ("La tua password è : <b>" & password & "</b>")
%>

```

TIPO: ASP

GENERARE UNA PASSWORD CASUALE (SOLO NUMERI)

```

<%
Randomize ' inizializza il generatore di numeri casuale
Do Until len(password) = 10 ' qui imposti la lunghezza della password
numero = Int((9 * Rnd) + 1) ' Genera un valore casuale compreso tra 1 e 9.
password = password & ; numero
Loop
Response.Write ("La tua password è : <b>" & password & "</b>")
%>

```

TIPO: ASP

INDIRIZZO IP

```

<%
function indirizzo()
indirizzo = "Il tuo indirizzo I.P. è : " & Request.ServerVariables("REMOTE_ADDR")
end function
response.write indirizzo()
%>

```

TIPO: ASP

LEGGERE I DATI RICEVUTI TRAMITE ?

```

<% dim risul

```

```
result=request.QueryString("errore")
response.write(result)
%>
```

TIPO: ASP

LEGGERE I PRIMI CARATTERI DI UNA STRINGA

```
<% dim testo
testo=left(testo,15)
response.wrtie(testo)
%>
```

TIPO: ASP

LEGGERE UNA VARIABILE DI SESSIONE

```
<% screenanme=session.contents.item("sname") %>
```

TIPO: ASP

OGGETTO OPEN

source, connection, cursortype, locktype, commandtype

CURSOR TYPE

dOpenDynamic 2 Apre un cursore dinamico
adOpenForwardOnly 0 Default. Apre un cursore "solo in avanti"
adOpenKeyset 1 Apre un cursore di tipo keyset
adOpenStatic 3 Apre un cursore di tipo statico
adOpenUnspecified -1 Indica un valore non specificato per il tipo di cursore

LOCKTYPE

adLockBatchOptimistic 4 I record non vengono bloccati finché non avviene un aggiornamento batch. Questa opzione dovrà essere utilizzata con cursori lato client e Recordset non connessi
adLockOptimistic 3 I record sono bloccati solo quando si chiama il metodo update
adLockPessimistic 2 I record sono bloccati immediatamente quando inizia la modifica per garantire che non vi siano discrepanze fra quando i nuovi valori sono stati inseriti e quando si verifica il comando Update
adLockReadOnly 1 Default. I record sono di sola lettura e non possono essere modificati. Inoltre, non si possono aggiungere nuovi record
adLockUnspecified -1 La copia dei dati viene creata con lo stesso tipo di bloccaggio dell'originale

TIPO: ASP

PAGINAZIONE DEI RISULTATI DI QUERY DI GRANDI DIMENSIONI

```
<%@LANGUAGE = VBScript%>
```

```
<%
```

```
' DEFINISCO LE VARIABILI CHE MI SERVONO PER L'APPLICAZIONE
```

```
Dim sc, cn, rs, quanti, pag, contatore
```

```
' DEFINISCO LA STRINGA DI CONNESSIONE
```

```
sc = ""
```

```
sc = sc & "driver={Microsoft Access Driver (*.mdb)};dbq="
```

```
sc = sc & Server.MapPath("database.mdb")
```

```
' IMPOSTO LA CONNESSIONE ED IL RECORDSET
```

```
Set cn = Server.CreateObject("ADODB.Connection")
```

```
Set rs = Server.CreateObject("ADODB.Recordset")
```

```
' APRO LA CONNESSIONE COL DATABASE
```

```
cn.Open sc
```

```
' DEFINISCO IL NUMERO DI RECORD DA VISUALIZZARE PER OGNI PAGINA
```

```
quanti = 5
```

```

' RECUPERO IL NUMERO DI PAGINA CORRENTE
' E CONTROLLO CHE NON SIA INFERIORE A 1
pag = Request.QueryString("pag")
If IsNumeric(pag) = False Or pag < 1 Then pag = 1

' IMPOSTO UN CONTATORE
contatore = 0
%>
<html>
<head>
<title>Paginazione dei risultati di query molto lunghe in ASP</title>
<style>a { color: blue; }</style>
</head>
<body>

<table align="center" width="300" border="1">
<%
' APRO IL RECORDSET
rs.Open "SELECT nome FROM utenti ORDER BY nome ASC", cn, 1
' EFFETTUA MATERIALMENTE LA PAGINAZIONE
rs.PageSize = quanti
rs.AbsolutePage = pag
' ESEGUO IL CICLO
While rs.EOF = False And contatore < quanti
%>
<tr><td><%=rs("nome")%></td></tr>
<%
rs.MoveNext
' INCREMENTO IL VALORE DEL CONTATORE
contatore = contatore + 1
Wend
%>
</table>

</body>
</html>
<%
' UN PO DI PULIZIA...
rs.Close
Set rs = Nothing
cn.Close
Set cn = Nothing
%>

```

TIPO: ASP

PAGINAZIONE II

```

<%
RecordsPerPage = 5
page = Request("page")
if page="" then page=1
' STRINGA SQL
strRicerca = Request("ricerca")
SQL = "SELECT * FROM Ricerca WHERE Descrizione LIKE '%" & strRicerca & '%"
'apri recordset
' paginazione
rs.PageSize = RecordsPerPage
rs.AbsolutePage = page
If rs.EOF=True or rs.BOF=True then
Response.Write "<P>Nessun risultato trovato</P>"
Else

```

```

For i=1 to RecordsPerPage
  if Not rs.EOF then
    'mostra record
    rs.MoveNext
  end if
Next
End if
%>Infine, per aggiungere una barra di navigazione tra le pagine con risultati:
<%
Response.Write "<P><B>Pagine:</B> "
For pag=1 to rs.PageCount
Response.Write "<A href='pagina.asp?page=" & pag
Response.Write "&" & Server.UrlEncode(strRicerca)
Response.write ">"
Response.Write pag
Response.Write "</A> "
Next
Response.Write "</P>"
%>

```

TIPO: ASP

PAGINAZIONE III

```

<!--la riga seguente equivale all'include di ADOVB.INC-->
<!-- METADATA TYPE="typelib" UUID="00000206-0000-0010-8000-00AA006D2EA4"
NAME="ADO Type Library" -->

```

```

<%

```

```

Set objRecordset = Server.CreateObject("ADODB.Recordset")
' modificare il path per puntare correttamente al database
strConnection = "Provider=Microsoft.Jet.Oledb.4.0;data source=" &
server.mappath("database.mdb")

' numero di record che vogliamo mostrare per pagina – utilizziamo una costante
CONST RECORDMAX = 10
' controlla se sia stato passato il parametro <page>, cioè il numero di pagina
If Request.QueryString("page").Count > 0 Then
  intPageNum = CInt(Request.QueryString("page"))
Else
  intPageNum = 1 'non è stato passato, di default iniziamo dalla prima pagina
End If
With objRecordset
  ' il seguente è il tipo di cursore più performante – non possiamo però utilizzare la proprietà
RecordCount (poco importa, non ci serve in questa sede...)
  .CursorType = adOpenForwardOnly
  ' la localizzazione del cursore di default è lato server, noi la spostiamo sul client
  .CursorLocation = adUseClient
  ' il tipo di Lock più performante
  .LockType = adLockReadOnly
  ' creiamo un buffer di memoria locale - dove verranno stipati i record - uguale a RECORDMAX
  .CacheSize = RECORDMAX
  .PageSize = RECORDMAX
  .Source = "SELECT * FROM tabella"
  .ActiveConnection = strConnection
  .Open ' eseguiamo la Query
  ' questo è importante: disconnettiamo il Recordset dalla sorgente dati (cioè il database).
  ' Uno dei maggiori vantaggi è dato dalla scalabilità: per 100 utenti che "scorrono" le pagine,
  ' non ci saranno altrettante connessioni e cursori aperti sul server. Per 1000 utenti idem.
  Set .ActiveConnection = Nothing
  intTotalPages = .PageCount
  If NOT .EOF then

```

```

' questo è importante: grazie al metodo GetRows dell'oggetto Recordset recuperiamo
' SOLTANTO il numero di record necessari per la pagina e li stipiamo in un array.
.AbsolutePage = intPageNum
arrData = .GetRows(RECORDMAX)
intColsTot = Ubound(arrData,1) ' numero di colonne, cioè di campi
intRowsTot = Ubound(arrData,2) ' numero di righe, cioè di records
End If
End With
' Rilasciamo le risorse
objRecordset.Close:Set objRecordset = Nothing

```

```

' Questo ciclo mostra il contenuto dei campi di ciascun record
' ovviamente va adattato e reso meno "spartano" (è qui solo per esempio)
For i = 0 To intRowsTot
  Response.Write "<BR>"
  For j = 0 To intColsTot
    Response.Write " "&arrData(j, i)
  Next
Next

```

```

'Pagina precedente – mostra una freccia a sinistra
Response.Write "<P>"
If intPageNum > 1 Then
  Response.Write( "<a href=""?page=" & intPageNum-1 &""><-- </a>" )
End If

```

```

'Mostra la pagina corrente in BOLD (non cliccabile) + tutte le altre (cliccabili)
For i = 1 to intTotalPages
  If i = intPageNum Then
    Response.Write(" <b>"&i"&"</b>")
  End If

```

TIPO: ASP

PAGINAZIONE MIA I

```

<%
Dim sc, cn, rs, quanti, pag, contatore
Dim Con
Set Con = Server.CreateObject("ADODB.Connection")
con.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" &
server.mappath("/quadernino/quadernino.mdb")

```

```

' -----
' DEFINISCO IL NUMERO DI RECORD DA VISUALIZZARE PER OGNI PAGINA
  quanti = 5
' RECUPERO IL NUMERO DI PAGINA CORRENTE
' E CONTROLLO CHE NON SIA INFERIORE A 1
  pag = Request.QueryString("pag")
  If IsNumeric(pag) = False Or pag < 1 Then pag = 1
' IMPOSTO UN CONTATORE
  contatore = 0
' -----

```

```

Dim rss
Set rss = Server.CreateObject("ADODB.Recordset")
rss.Open "select * from script order by id desc", con ,1,3

```

```

rss.PageSize = quanti
rss.AbsolutePage = pag
%>
</font></p>
  <%response.write("<table border=0 align=center cellpadding=0 cellspacing=0>")%></th>
</tr>

```

```

<tr>
  <td ><span class="style48">TIPO</span></td>
  <td ><span class="style48">TITOLO</span></td>
  <td><span class="style48"> </span></td>
  <td align="center" ><span class="style48">COMMENTI</span></td>
</tr>
<tr>
  <td ></td>
  <td ></td>
  <td ></td>
  <% Do while NOT RSs.EOF and contatore < quanti %>
  <%Dim id
id=rss("id")
  Dim titolo
titolo=rss("tipo")
dim autore
autore=rss("titolo")
'dim quando
'quando=rss("data")
'dim commenti
'commenti=rss("commenti")

%>
</tr>
<tr>
  <td valign="top"><%response.write("<a href=leggis.asp?idd=")%>
  <%response.write(id)%>
  <%response.write(">")%>
  <%response.write("LEGGI")%>
  <%response.write("</a>")%></td>
  <td valign="top">
  <%response.write("<a href=leggis.asp?idd=")%>
  <%response.write(id)%>
  <%response.write(">")%>
<span style="text-transform: capitalize">
  <%response.write(titolo)%></span>
  <%response.write("</a>")%> </td>
  <td valign="top" ><%response.write("<a href=leggis.asp?idd=")%>
  <%response.write(id)%>
  <%response.write(">")%>
  <%response.write(autore)%>

```

TIPO: ASP

QUANTI RECORD RESTITUISCE UNA QUERY

```

<%
Dim cn,cm
Set cn = Server.CreateObject("ADODB.Connection")
cn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" & Server.MapPath("Archivio.mdb")
set rs=cn.execute ("Select count(*) as quanti from tabella")
response.write "Record nel database: "&rs("quanti")
cn.close
Set cn = NOTHING
Set rs = Nothing
%>

```

TIPO: ASP

RICAVARE UN NUMERO UTENTE UNIVOCO

```

<%
numero_cliente=Session.SessionID

```

```
Response.Write "Ti è stato assegnato il codice "&numero_cliente&"."  
>
```

TIPO: ASP

SCRIVERE DATI IN UN DB

```
<%  
  
set conn = server.createobject("adodb.connection")  
conn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" &  
server.mappath("/utenti/messaggi.mdb")  
set rs = server.createobject("adodb.recordset")  
rs.open "utenti", conn, 2, 2  
rs.addnew  
rs("userepass") = namepass  
rs("screename") = screename  
rs("note") = note  
rs("email") = email  
rs("webpage") = website  
rs.update  
rs.movelast  
'strID = rs("ID")  
rs.close  
set rs= nothing  
set conn = nothing  
  
>
```

TIPO: ASP

SCRIVERE UNA VARIABILE DI SESSIONE CHE POSSO RIUTILIZZARE IN ALTRE PAGINE

```
<% session.contents.item("sname")= sn %>
```

TIPO: ASP

SESSION END

```
<html>  
<head>  
<script language="vbscript" runat="server">  
Sub Session_OnStart  
Operazioni all'inizio di sessione  
End Sub  
  
Sub Session_OnEnd  
Operazioni alla fine della sessione  
End Sub  
</script>  
</html>  
</head>
```

TIPO: ASP

TIPI DI CURSORE E DI BLOCCHI

```
strSQL = "SELECT * FROM nometabella"  
objRs.Open objConn, strSQL, 1, 2  
Analizzando la seconda riga, troviamo  
objRs.Open : il comando di apertura del recordset  
objConn : il parametro che definisce la connessione  
strSQL : il parametro che definisce la query SQL da eseguire  
1 : il parametro che definisce il CursorType  
2 : il parametro che definisce il LockType  
CursorType  
CursorType definisce il tipo di cursore di un recordset.  
Per comprendere meglio la sua funzione immaginiamo per un attimo di accedere ad una tabella di  
un database Access attraverso la console di gestione. Non appena aperta la tabella noteremo un
```

puntatore, gestibile attraverso la tastiera ed il mouse, posizionarsi sulla prima cella in alto a sinistra.

Questo cursore definisce il record (ovvero la riga) e la cella in cui ci troviamo e nella quale si rifletteranno eventuali comandi di aggiornamento o modifica.

L'utilità del cursore nel caso di una connessione al database tramite ASP è la stessa. Ci consente di navigare attraverso i record e di eseguire operazioni su di essi.

Tipi di Cursore

Esistono 4 tipi di cursori, ognuno con le proprie caratteristiche.

Forward Only(solo in avanti)

È il cursore predefinito quando non è specificato un valore alternativo. Consente di muoversi solo in avanti attraverso i record. La costante ADO che lo identifica è `adOpenForwardOnly` mentre il valore è 0.

Static(Statico)

Permette di spostarsi avanti ed indietro tra i record ma non riflette le modifiche eseguite da altri utenti. La costante ADO che lo identifica è `adOpenStatic` mentre il valore è 3.

Dynamic(Dinamico)

È il cursore più flessibile. Consente di muoversi avanti ed indietro e riflette le modifiche eseguite da altri utenti. La costante ADO che lo identifica è `adOpenDynamic` mentre il valore è 2.

Keyset(Set di tasti)

Permette lo spostamento avanti ed indietro, riflette le modifiche eseguite da altri utenti ad eccezione dell'inserimento e della rimozione di record. La costante ADO che lo identifica è `adOpenKeyset` mentre il valore è 1.

Impostazione del Cursore

Per impostare il tipo di cursore è possibile procedere in due modi: la proprietà `.CursorType` prima di aprire il recordset oppure la specifica del parametro in fase di esecuzione della query.

In entrambi i casi è possibile inserire il valore sia come costante ADO sia, scelta che raccomando, come valore numerico.

Nella logica di programmazione l'uso di una costante è più opportuna poiché rende più intuitiva la comprensione del codice. Tuttavia, ritengo inutile in questo caso imparare a memoria lunghi nomi di costanti per la specifica di un parametro standard.

Il primo metodo, assumendo che `objRs` sia il nostro recordset, sarà

```
objRs.CursorType = 1 ' oppure la costante
```

mentre il secondo

```
objRs.Open objConn, strSQL, 1 ' oppure la costante
```

Conclusioni

Arrivati a questo punto molti potrebbero chiedersi lo scopo o l'utilità di comprendere l'utilizzo dei cursori.

Come anticipato, definire il cursore corretto è molto importante per due ragioni. In primo luogo consente di risparmiare risorse, in secondo luogo un cursore errato non permette di completare l'esecuzione di un comando.

Un cursore come Forward Only non permette, ad esempio, di eseguire la paginazione dei record.

Non solo, un cursore predefinito è anche la risposta al perché la proprietà `.RecordCount` restituisce -1 invece del reale numero di record. In questo caso è necessario impostare il cursore a 1.

LockType

LockType definisce il tipo di blocco per l'aggiornamento di un recordset.

Anche in questo caso possiamo comprendere meglio con un esempio. Riprendiamo il database Access di prima.

Fin tanto che sono io l'unico utilizzatore non ci saranno mai problemi di sovrascrittura dei dati. Ma cosa succederebbe se due persone tentassero di aggiornare contemporaneamente lo stesso dato? Quale dei due sarebbe il valore finale?

Per risolvere questo problema ADO mette a disposizione una serie di valori per consentire di attivare un blocco quando si eseguono modifiche su un recordset.

Tipi di Blocco

Esistono 4 tipologie differenti di blocco.

Read Only (Sola lettura)

TIPO: ASP

USARE LE FORM IN ASP

```
<html>
```

```
<body>
```

```
<form action="demo_requery.asp" method="get">
```

```

Your name: <input type="text" name="fname" size="20">
<input type="submit" value="Submit">
</form>
<% dim fname
fname=Request.QueryString("fname")
If fname<>" " Then
    Response.Write("Hello " & fname & "!<br />")
    Response.Write("How are you today?")
End If %>
</body>
</html>

```

TIPO: ASP

UTENTE CONNESSO O NON CONNESSO

```

<%
dim connessione
connessione=Response.IsClientConnected
response.write(connessione)
%>

```

TIPO: ASP

VERIFICARE L'ESISTENZA DI UN FILE

```

<%OPTION EXPLICIT
' questo codice verifica l'esistenza di un file
dim fso
dim file
set fso = Server.CreateObject("Scripting.FileSystemObject")
file = server.mappath("nomecartella/nomefile.ext")
if fso.fileExists(file) then
response.write "Il file "& file & " esiste!"
else
response.write "Il file " & file & " non esiste!"
end if
set fso = nothing
%>

```

TIPO: ASP , JAVA

STAMPARE UNA PAGINA CON JAVASCRIPT

```

<A HREF="javascript:window.print()"> STAMPA </A>
oppure con un form
<FORM>
<INPUT TYPE="button" onClick="window.print()">
</FORM>

```

TIPO: ASP E ASP.NET

SAMPLE UPLOAD CON SMART UPLOAD

```

<HTML>
<BODY BGCOLOR="white">

```

```

<H1>aspSmartUpload : Sample 1</H1>
<HR>

```

```

<FORM METHOD="POST" ACTION="/scripts/aspSmartUpload/Sample1.asp"
ENCTYPE="multipart/form-data">
    <INPUT TYPE="FILE" NAME="FILE1" SIZE="50"><BR>
    <INPUT TYPE="FILE" NAME="FILE2" SIZE="50"><BR>
    <INPUT TYPE="FILE" NAME="FILE3" SIZE="50"><BR>
    <INPUT TYPE="FILE" NAME="FILE4" SIZE="50"><BR>
    <INPUT TYPE="SUBMIT" VALUE="Upload">
</FORM>

```

```

</BODY>
</HTML>
<HTML>
<BODY BGCOLOR="white">

<H1>aspSmartUpload : Sample 1</H1>
<HR>

<%
' Variables
' *****
    Dim mySmartUpload
    Dim intCount

' Object creation
' *****
    Set mySmartUpload = Server.CreateObject("aspSmartUpload.SmartUpload")

' Upload
' *****
    mySmartUpload.Upload

' Save the files with their original names in a virtual path of the web server
' *****
    intCount = mySmartUpload.Save("/aspSmartUpload/Upload")
' sample with a physical path
' intCount = mySmartUpload.Save("c:\temp")

' Display the number of files uploaded
' *****
    Response.Write(intCount & " file(s) uploaded.")
%>
</BODY>
</HTML>

```

TIPO: ASP NET

COLLEGARSI AD ACCESS (1)

```

<SCRIPT language="vb" runat="server">
Sub Page_Load(Sender As Object, E as EventArgs)

    Dim connessione As New System.Data.OleDb.OleDbConnection()
    Dim sql As New System.Data.OleDb.OleDbCommand()
    connessione.ConnectionString = "provider=Microsoft.Jet.OLEDB.4.0; "
    &"DataSource=" &server.mappat(/dati/database.mdb;")
    sql.CommandText = "SELECT * FROM nome_tabella"
    sql.Connection = connessione

    connessione.Open()

    risultati.DataSource= sql.ExecuteReader()
    risultati.DataBind()
    sql.Connection.Close()

end sub
</SCRIPT>
<asp:DataGrid id="risultati " runat="server" />

```

TIPO: ASP NET

CONNESSIONE AD UN DATABASE

```

<SCRIPT language="vb" runat="server">
Sub Page_Load(Sender As Object, E as EventArgs)
  Dim connessione As New System.Data.OleDb.OleDbConnection()
  Dim sql As New System.Data.OleDb.OleDbCommand()
  connessione.ConnectionString = "provider=Microsoft.Jet.OLEDB.4.0; " & _
"Data Source=C:\inetpub\wwwroot\database.mdb;"
  sql.CommandText = "SELECT * FROM nome_tabella"
  sql.Connection = connessione

  connessione.Open()
  risultati.DataSource= sql.ExecuteReader()
  risultati.DataBind()
  sql.Connection.Close()
end sub
</SCRIPT>
<asp:DataGrid id="risultati " runat="server" />

```

TIPO: ASP NET

CONNESSIONE AD UN DB ACCESS

```

<%@ Page Language = "VB" debug = "True"%>
<%@ Import NameSpace = "System.Data.OleDb" %>

<script runat = "Server">
sub Page_Load(sender as Object, e as EventArgs)
'Impostiamo la stringa di connessione as Database
dim strConn as String
strConn = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.MapPath("DB.mdb")

'Creiamo l' oggetto objCon
dim objCon as OleDbConnection = new OleDbConnection(strConn)

'Apriamo la connessione al Database
objCon.Open

'Chiudiamo la connessione
objCon.Close
end sub
</script>

<html>
<head>
  <title>Connessione DataBase con ASP.Net</title>
</head>

<body>

</body>
</html>

```

TIPO: ASP NET

VILIZZARE LE IMMAGINI FACENDO INTERAGIRE VB.NET CON L'HTML.

```

<%@page language="vb" debug="true"%>
<script runat="server">
Sub page_load(obj As object, e As EventArgs )
immagine.src ="immagine.jpg"
End sub
</script>
<img id="immagine" runat="server" />

```

TIPO: ASP NET

LEGGERE I DATI DA UN DATABASE

```

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

<script runat="server">
sub Page_Load
dim dbconn,sql,dbcomm,dbread
dbconn=New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;data source=" &
server.mappath("/fdati/daba.mdb"))
dbconn.Open()
sql="SELECT * FROM customers"
dbcomm=New OleDbCommand(sql,dbconn)
dbread=dbcomm.ExecuteReader()
customers.DataSource=dbread
customers.DataBind()
dbread.Close()
dbconn.Close()
end sub
</script>

<html>
<body>

<form runat="server">
<asp:Repeater id="customers" runat="server">

<HeaderTemplate>
<table border="1" width="100%">
<tr bgcolor="#b0c4de">
<th>nome</th>
<th>messaggi</th>
<th>file</th>
</tr>
</HeaderTemplate>

<ItemTemplate>
<tr bgcolor="#f0f0f0">
<td><%#Container.DataItem("nome")%> </td>
<td><%#Container.DataItem("messaggi")%> </td>
<td><%#Container.DataItem("file")%> </td>
</tr>
</ItemTemplate>

<FooterTemplate>
</table>
</FooterTemplate>

</asp:Repeater>
</form>

</body>
</html>

```

TIPO: ASP NET

VALIDARE UN FORM

```

<%@ page language="vb" %>
<%@ import Namespace="System.Web.UI.WebControls" %>
<script runat="server">
Sub elabora( obj As Object , e As EventArgs )
If page.IsValid then
lbMsg.text = " Modulo compilato correttamente"
End If
End Sub

```

```
</script>
<form runat="server">
Campo obbligatorio
<asp:TextBox id="tbObb" runat="server" />
<br>
Campo <b>NON</b> obbligatorio
<asp:TextBox id="tbLibero" runat="server" />
<br>
<asp:Button id="btSubmit" runat="server" text="Prova" onClick="Elabora" />
<br>
<asp:RequiredFieldValidator runat="server" controlToValidate="tbObb" errorMessage="Immettere
campo obbligatorio" />
<br>
<asp:Label id="lbMsg" value="" runat="server" />
</form>
```

TIPO: ASP, ASP.NET, HTML

REINDIRIZZARE UN DATO AD UNA PAGINA TRAMITE ?

```
<% response.write("<a href=click.asp?errore=") %>
    <% response.write(data) %>
    <% response.write(">CANCELLA</a>") %>
```

TIPO: ASP, ASP NET

CONVERTIRE LA DATA NEL FORMATO ITALIANO

```
<%
dim itdate
itdate = Day(data)&"/"&Month(data)&"/"&Year(data)
%>
```

TIPO: ASP.NET

CHIDE IL NOME E LO VISUALIZZA SENZA PASSARE VARIABILI AD UN'ALTRA PAGINA

```
<html>
  <script language="VB" runat="server">
    Sub EnterBtn_Click(Sender As Object, E As EventArgs)
      Message.Text = "Hi " & Name.Text & ", welcome to ASP.NET!"
    End Sub
  </script>
  <body>
    <h3><font face="Verdana">Handling Control Action Events</font></h3>
    <p>
      This sample demonstrates how to access a &lt;asp:textbox&gt; server control within the "Click"
      event of a &lt;asp:button&gt;, and use its content to modify the text of a &lt;asp:label&gt;.
    <p>
    <hr>
    <form action="controls3.aspx" runat="server">
      <font face="Verdana">
        Please enter your name: <asp:textbox id="Name" runat="server"/>
          <asp:button text="Enter" Onclick="EnterBtn_Click" runat="server"/>
      <p>
        <asp:label id="Message" runat="server"/>
      </font>
    </form>
  </body>
</html>
```

TIPO: ASP.NET

COLLEGARSI AD UN DATABASE CON ASP.NET

```
<%@ Page language="VB" Debug="false" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>
<script language="VB" runat="server">
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
  Dim dbConn As OleDbConnection
  Dim myAdapter As OleDbDataAdapter
  Dim sqlCmd As String
  Dim strConn As String
  Dim oDataSet As New DataSet

  strConn = ""
  strConn = strConn & "Provider=Microsoft.Jet.OLEDB.4.0;"
  strConn = strConn & "Data Source=" & Server.MapPath("database/utenti.mdb") & ";"
  strConn = strConn & "Persist Security Info=False"

  sqlCmd = ""
  sqlCmd = sqlCmd & "SELECT "
  sqlCmd = sqlCmd & " nome, "
  sqlCmd = sqlCmd & " cognome, "
  sqlCmd = sqlCmd & " email "
  sqlCmd = sqlCmd & " FROM utenti"

  dbConn = New OleDbConnection(strConn)
  myAdapter = New OleDbDataAdapter(sqlCmd, dbConn)
  myAdapter.Fill(oDataSet, "utenti")

  myDatagrid.DataSource = oDataSet.Tables("utenti").DefaultView
  myDatagrid.DataBind()
```

```
End Sub
</script>
<html>
<body>
<asp:DataGrid id="myDatagrid" runat="server" />
</body>
</html>
```

TIPO: ASP.NET

INSERIRE DEI DATI IN UNA TABELLA

```
<%@ Page Language="VB" Debug="true" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SqlClient" %>
<script language="VB" runat="server">
Sub Page_Load(Source as Object, E as EventArgs)
  Dim cnn As New SqlConnection("server=DBSERVER;User
id=USER;password=PASSWORD;database=DB")
  Dim InsertCommand As SqlCommand = New SqlCommand()
  InsertCommand.Connection = cnn
  Dim sql As String
  sql = "INSERT INTO TABELLA (campo) VALUES ('valore')"
  InsertCommand.CommandText = sql
  Try
    cnn.Open()
    InsertCommand.ExecuteNonQuery()
  Catch ex As Exception
    response.Write(ex.ToString())
  Finally
    cnn.Close()
  End Try
end sub
</script>
```