28/10/2013

Informatica per la Finanza

Informatica per le Scienze Economiche

Informatica e Programmazione per il Turismo

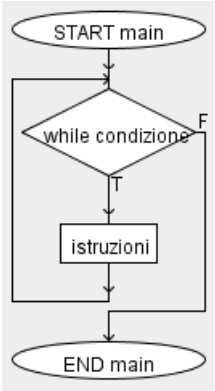
Istruzioni Cicliche

- VBA presenta diversi tipi di istruzioni cicliche alcune con condizione di permanenza (**`While`**) altre con condizioni di uscita (**`Until`**) dal ciclo.
  - Per evitare complicazioni si presenteranno solo quelle con condizione di permanenza nel ciclo

Istruzioni Cicliche

Do While *condizione*
  Istruzioni
Loop

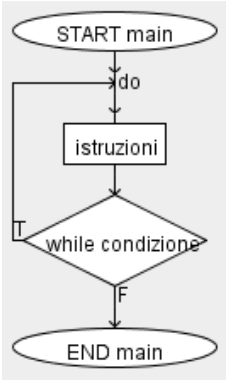
- Può non eseguire mai le istruzioni nel ciclo
- Per uscire da questo ciclo prima che la condizione sia verificata si usa l'istruzione
  Exit Do



Istruzioni Cicliche

Do
  Istruzioni
Loop While con*dizione*

- Esegue almeno una volta le istruzioni nel ciclo
- Per uscire da questo ciclo prima che la condizione sia verificata si usa l'istruzione
  Exit Do



Istruzioni Cicliche

- Un'alternativa all'istruzione **`Do While`** è l'istruzione **`While`**:

```
While condizione
    istruzioni
Wend
```

  – ha un comportamento identico a **`Do While`** ma non permette l'uscita anticipata dal ciclo

Istruzioni Cicliche

```
Sub prova()
    Dim riga As Integer

    riga = 1
    Cells(riga, 2) = Range("A1")
    Do While (Cells(riga, 2) > 0)
       Cells(riga + 1, 2) = Cells(riga, 2) - 1
       riga = riga + 1
    Loop

    riga = 1
    Cells(riga, 3) = Range("A1")
    Do
       Cells(riga + 1, 3) = Cells(riga, 3) - 1
       riga = riga + 1
    Loop While (Cells(riga, 3) > 0)

End Sub
```

- Provare con in A1
  - **0**
  - **8**
  - **-1**

Istruzioni cicliche

```
Option Explicit
Sub provaPre()
  Dim DoPre As Integer
  Dim par1 As Integer

  par1 = 10
  DoPre = 0
  Do While par1 > 10
    DoPre = DoPre + par1
    par1 = par1 - 1
  Loop
  Range("A1") = par1
  Range("A2") = DoPre
End Sub
```

```
Risultato:
10
0
```

```
Option Explicit
Sub provaPost()
  Dim DoPos As Integer
  Dim par2 As Integer

  par2 = 10
  DoPos = 0
  Do
    DoPos = DoPos + par2
    par2 = par2 - 1
  Loop While par2 > 10
  Range("B1") = par2
  Range("B2") = DoPos
End Sub
```

```
Risultato:
9
10
```


Istruzioni Cicliche

- Provando **`provaPre()`**
  - Se **`par1 = 5`** allora **`DoPre = 0`**
    - Il ciclo non viene eseguito
  - Se **`par1 = 20`** allora **`DoPre = 155`**
- Provando **`provaPost()`**
  - Se **`par2 = 5`** allora **`DoPre = 5`**
    - Il ciclo è sempre eseguito almeno una volta
  - Se **`par2 = 20`** allora **`DoPre = 155`**

Istruzioni Cicliche

```
Option Explicit
Sub provaPre()
  Dim DoPre As Integer
  Dim par1 As Integer

  par1 = 15
  DoPre = 0
  Do While par1 > 10
    DoPre = DoPre + par1
    par1 = par1 - 1
  Loop
  Range("A1") = par1
  Range("A2") = DoPre
End Sub
```

DoPre	par1
0	15
15	15
15	14
29	14
29	13
42	13
42	12
54	12
54	11
65	11
65	10

Istruzioni Cicliche

```
Option Explicit
Sub provaPost()
  Dim DoPos As Integer
  Dim par2 As Integer

  par2 = 15
  DoPos = 0
  Do
    DoPos = DoPos + par2
    par2 = par2 - 1
  Loop While par2 > 10
  Range("B1") = par2
  Range("B2") = DoPos
End Sub
```

DoPos	par2
0	15
15	15
15	14
29	14
29	13
42	13
42	12
54	12
54	11
65	11
65	10

In questo caso il comportamento è identico nei due modi

Istruzioni Cicliche

```
'verificare quanti numeri ci sono nella colonna A
'del foglio di lavoro considerando che i valori
'iniziano dalla cella A1 e terminano appena
' si incontra una cella vuota
Option Explicit
Sub prova()
  Dim i As Integer, conta

  i = 1: conta = 0
  Do While (Not IsEmpty(Cells(i, 1)))
    If (IsNumeric(Cells(i, 1))) Then
     conta = conta + 1
    End If
    i = i + 1
  Loop

  Cells(1, 2) = conta
End Sub
```

	A	B
1	12	6
2	-36	
3	14/10/2013	
4	lkdfklfdh	
5	125	
6	36	
7	VERO	
8	12,36	
9	-1,00E+12	

`IsEmpty()` verifica se la cella argomento è vuota
`IsNumeric()` verifica se la cella argomento contiene un numero; anche i valori **`Boolean`** sono considerati numeri

Istruzioni Cicliche

- sommare tutti valori nella colonna A finché non si incontra una cella vuota nella colonna e visualizzare il risultato con MsgBox (supponiamo che ci siano solo numeri)
  - *finché non incontra una cella vuota*
    - Osservare che la condizione di uscita è indicata al contrario
      - Usare questa condizione negandola (con **`Not`**)
      - Calcolare l'opposto della condizione

Istruzioni Cicliche

```
Option Explicit
Sub prv()
  Dim riga As Integer
  Dim tot As Double

  riga = 1

  Do While (Not IsEmpty(Cells(riga, 1)))
     tot = tot + Cells(riga, 1)
     riga = riga + 1
  Loop

  Range("B1") = tot

End Sub
```


Istruzioni Cicliche

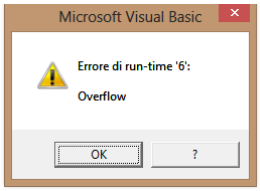
```
Option Explicit
Sub prova()
    Dim v As Integer
    v = 4

    Do While True
      v = v + 1
      Range("F1") = v
    Loop

    Range("D1") = v
End Sub
```

- Questo esempio mostra cosa succede quando si supera il valore più grande rappresentabile **`Integer`**
  - Notare il comportamento diverso dal C/C++

v
4
5
6
:
:
32767



Overflow

Operatori Booleani

Not (negazione) operatore unario
And (E logico) operatore binario
Or (O logica) operatore binario

A	B	Not A	A And B	A Or B
V	V	F	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	F

Istruzioni Cicliche

```
Sub prova()
    Dim v As Integer, q As Integer

    v = 4
    q = 5
    Do While v > 2 Or q < 5
      v = v - 1
      q = q - 1
    Loop

    Range("D1") = v
    Range("D2") = q
End Sub
```

v	q
4	5
3	5
3	4
2	4
2	3
1	3
1	2
0	2
0	1
-1	1
-1	0
-2	0
-2	-1

∞

Istruzioni Cicliche

```
Sub prova()
    Dim v As Integer, q As Integer

    v = 4
    q = 5
    Do While v > 2 AND q < 5
      v = v - 1
      q = q - 1
    Loop

    Range("D1") = v
    Range("D2") = q
End Sub
```

```
v     q
4     5

NON ENTRA NEL CICLO
```


Istruzioni Cicliche

- chiedere all'utente un numero intero positivo e scriverlo in A1
  - calcolare il fattoriale del valore in A1 e scriverlo in A2

Istruzioni Cicliche

```
Option Explicit
Sub prova()
    Dim riga As Integer, v As String
    Dim fat As Integer, n As Integer

    'acquisizione numero intero positivo
    Do
       v = InputBox("dammi un intero positivo: ")
       If IsNumeric(v) Then
         If CInt(v) = v Then
            If v > 0 Then
               Exit Do
            End If
          End If
       End If
    Loop While (True)

    Range("a1") = v

    'calcolo fattoriale
    n = CInt(v)
    fat = 1
    Do While (n > 1)
      fat = fat * n
      n = n - 1
    Loop

    Range("a2") = fat
End Sub
```

_ permette di andare a capo (è preceduto da uno spazio)

In alternativa:

```
Do
   v = InputBox("dammi un intero positivo: ")
Loop While Not (IsNumeric(v) _
                And (CInt(v) = v) And CInt(v) > 0)
```

Questa soluzione ha il problema che leggendo 3.9 viene considerato 39 da **`Cint()`** ed il confronto `CInt(v) = v` risulta vero; invece 3,9 viene arrotondato da **`Cint()`** in 4.
Il problema può essere risolto usando le espressioni regolari

Istruzioni Cicliche

```
'verificare quante celle non vuote ci sono nella colonna A del foglio di lavoro
'Il conteggio inizia dopo la prima cella non vuota  e termina appena
'si incontra una cella vuota
Sub prova()
  Dim i As Integer, conta As Integer


  i = 1
  Do While (IsEmpty(Cells(i, 1)))
    i = i + 1
  Loop

  conta = 1 'sono uscito dal precedente perche'
            'ho trovato una cella non vuota
  Do While (Not IsEmpty(Cells(i, 1)))
     i = i + 1
     conta = conta + 1
  Loop

  Cells(1, 2) = conta
  Cells(2, 3) = i
End Sub
```

Istruzioni Cicliche

- Richiedere all'utente un numero finché questo non è pari

```
Option Explicit
Sub provaPre()
  Dim val As Integer

  Do
    val = InputBox("dammi un intero pari: ")
  Loop While (val Mod 2 <> 0)

  Range("H2") = val
End Sub

OPPURE

Option Explicit
Sub provaPre()
  Dim val As Integer

  val = 7   'FORZO INGRESSO
  Do While (val Mod 2 <> 0)
    val = InputBox("dammi un intero pari: ")
  Loop

  Range("H2") = val
End Sub
```

Se il valore inserito
Non è di tipo numerico
Ci sono dei problemi

Istruzioni Cicliche

```
Option Explicit
'leggere un numero finché l'utente non inserisce un numero pari
Sub prova()
  Dim val As Integer, resto As Integer, v As Variant


  Do
     v = InputBox("dammi un intero pari: ", "Inserimento valore")
     If (IsNumeric(v)) Then
       resto = CInt(v) Mod 2
     Else
        MsgBox ("Birichino: voglio un numero")
        resto = 1
     End If
  Loop While (resto <> 0)

End Sub
```

Istruzioni Cicliche

```
For cont = valIni To valFin Step passo
  Istruzioni
Next
```

Dove

- ***cont*** è una variabile numerica
- ***valIni*** è il valore iniziale di ***cont***
- ***valFin*** è il valore finale
- ***passo*** è la quantità di cui incrementare ad ogni ciclo
  - Se **Step *passo*** è omesso ***passo*** vale 1

- Per uscire dal ciclo anzitempo si usa ***Exit For***

Istruzioni Cicliche

```
For cont = valIni To valFin Step passo
  Istruzioni
Next
```

Equivale a:

```
cont = valIni
Do While (cont <= valFin)
    Istruzioni
    cont = cont + passo
Loop
```

Istruzioni Cicliche

```
Option Explicit
Sub provaFor()
  Dim val As Double, i As Integer

  i = 1
  For val = 10.5 To 5.2 Step -0.8
      Cells(i, 7) = val
      i = i + 1
  Next
End Sub
```


Istruzioni Cicliche

$$s = \sum_{r=1}^{5} (r) = 1+2+3+4+5$$

s	r		
0	1		s=0
1	1	s=s+r	r=1
1	2	r=r+1	Do while (r <= 5)
			s=s+r
3	2	s=s+r	r=r+1
3	3	r=r+1	Loop
6	3	s=s+r	s=0
6	4	r=r+1	for r=1 to 5
10	4	s=s+r	s=s+r
10	5	r=r+1	Next
15	5	s=s+r	
15	6	r=r+1	

L'istruzione for è molto utile per gestire le sommatorie e le produttorie

Istruzioni Cicliche

```
Option Explicit
Sub prova()
    Dim i As Integer, s As Integer

    s = 0
    For i = 1 To 5 Step 1
      s = s + i
    Next

    Range("a1") = s
End Sub
```


Istruzioni Cicliche

$$s = \sum_{r=1}^{5} \left(\prod_{k=r}^{6} \right) = (1*2*3*4*5*6) + (2*3*4*5*6) + (3*4*5*6)$$

$$\ldots + (4*5*6) + (5*6)$$

```
Option Explicit
Sub prova()
    Dim r As Integer, k As Integer
    Dim s As Integer, p As Integer

    s = 0
    For r = 1 To 5 Step 1

      p = 1
      For k = r To 6
        p = p * k
      Next

      s = s + p
    Next

    Range("a1") = s
End Sub
```

Istruzioni cicliche

- leggere il valore nella cella E1 se è un numero intero positivo calcolarne il fattoriale
- e scrivere il valore nella cella E2

```
Option Explicit

Private Sub Lavora ()
    Dim fatt As Long, i As Integer

    If Not IsEmpty(Range("E1")) Then
      If Range("E1") >= 0 Then
         If CInt(Range("E1").Value) = Range("E1") Then
            fatt = 1 ' altrimenti vale 0
            For i = 1 To Range("E1").Value
              fatt = fatt * i
            Next
            Range("E2") = fatt
         End If
       End If
    End If
End Sub
```


Istruzioni Cicliche

- Leggere dal foglio di lavoro tutti i valori compresi nell'intervallo A1..D27 e sommare fra loro i valori numerici e contare quanti valori non numerici si sono incontrati e quante celle vuote si sono incontrate.
- I valori trovati vanno messi rispettivamente in E1, E2 ed E3
- La funzione `IsNumeric()` restituisce True se il valore della cella ha un formato numerico

```
Option Explicit

Private Sub Lavora_Click()
    Dim riga As Integer, col As Integer
    Dim somma As Double

    somma = 0
    Range("E2") = 0
    Range("E3") = 0

    For riga = 1 To 27
      For col = 1 To 4
        If IsEmpty(Cells(riga, col)) Then
           Range("E3") = Range("E3") + 1
         ElseIf IsNumeric(Cells(riga, col)) Then
           somma = somma + Cells(riga, col)
         Else
           Range("E2") = Range("E2") + 1
         End If
      Next
    Next

    Range("E1") = somma
End Sub
```

Istruzioni Cicliche

- Per gestire un ciclo su di un intervallo di valori si può usare:

```
For Each elemento In Gruppo
  istruzioni
Next
```

- dove
  - ***Gruppo*** è un insieme di celle; si può indicare usando **Range(*cellaSupSx,cellaInfDx*)**
- per uscire dal ciclo prima del tempo si può usare **Exit For**

Istruzioni cicliche

```
Sub prv2()
  Dim r As Range
  For Each r In
  Range("A1", "C3")
   MsgBox (r.Row & _
    "    " & r.Column)
  Next
End Sub
```

```
Sub prv3()
  Dim r As Range
  For Each r In
  Range("A1:C3")
   MsgBox (r.Row & _
    "    " & r.Column)
  Next
End Sub
```

Due modi equivalenti di indicare l'intervallo

Istruzioni Cicliche

- Sommare tutti i valori di tipo numerico nelle celle da A1 ad F6

```
Option Explicit
Sub EsempioForEach()

 Dim x As Variant, som As Double


 For Each x In Range("A1", "F6")
      If (IsNumeric(x.Value)) Then
        som = som + x.Value
      End If
 Next

 Range("H3") = som
End Sub
```


Istruzioni Cicliche

- Leggere dal foglio di lavoro tutti i valori compresi nell'intervallo A1..D27 e sommare fra loro i valori numerici e contare quanti valori non numerici si sono incontrati e quante celle vuote si sono incontrate.
- I valori trovati vanno messi rispettivamente in E1, E2 ed E3

```
Option Explicit

Private Sub Lavora_Click()
    Dim el As Variant
    Dim somma As Double

    somma = 0
    Range("E2") = 0
    Range("E3") = 0

    For Each el In Range("A1:D27")
        If IsEmpty(el) Then
           Range("E3") = Range("E3") + 1
         ElseIf IsNumeric(el) Then
           somma = somma + el.Value
         Else
           Range("E2") = Range("E2") + 1
         End If
    Next

    Range("E1") = somma
End Sub
```

Istruzioni Cicliche

- Calcolare la somma di tutti gli elementi compresi fra le celle A1 e D5

```
Sub esempioForEach()
 Dim r As Range
 Dim tot As Double

 tot = 0
 For Each r In Range("A1", "D5")
     tot = tot + r.Value
 Next

   Range("E1") = tot
End Sub
```


Istruzioni Cicliche

- Calcolare la somma di tutti i valori compresi fra A1 e D27 e fra F1 e G10

```
Option Explicit

Private Sub Lavora ()
    Dim el As Variant
    Dim somma As Double

    somma = 0
    For Each el In Range("A1:D27")
       If IsNumeric(el.Value) Then
           somma = somma + el.Value
         End If
    Next
    For Each el In Range("F1:G10")
       If IsNumeric(el.Value) Then
           somma = somma + el.Value
       End If
    Next
    Range("E1") = somma
End Sub
```

Istruzioni Cicliche

- Calcolare la somma di tutti i valori compresi fra A1 e D27 e fra F1 e G10

```
Private Sub Lavora2()
    Dim el As Variant
    Dim somma As Double

    somma = 0
    For Each el In Range("A1:D27", "F1:G10")
       If IsNumeric(el.Value) Then
           somma = somma + el.Value
       End If
    Next
    Range("E2") = somma
End Sub
```

Attenzione in questo modo si calcola la somma da A1 a G27!

Esercizio

- determina il valore minimo ed il valore massimo dei valori numerici presenti fra A1 e D27
- mette i risultati a video con una MsgBox

```
Option Explicit

Private Sub Lavora_Click()
    Dim el As Variant
    Dim min As Double, max As Double

    min = Range("A1").Value
    max = Range("A1").Value
    For Each el In Range("A1:D27")
      If (Not IsEmpty(el)) Then
        If (el.Value > max) Then
          max = el.Value
        End If

        If (el.Value < min) Then
          min = el.Value
        End If
      End If
    Next

    MsgBox ("max = " & max & " min = " & min)
End Sub
```

Istruzioni Cicliche

- Sommare tutti i valori fra par1 e par2 con un incremento di 1.5

Osservare che ct è di tipo Double

```
Sub EsempioFor()

  Dim Par1 As Double
  Dim Par2 As Double
  Dim tp As Double
  Dim  tot As Double
  Dim ct As Double

  tot = 0
  Par1 = Range("A1").Value
  Par2 = Range("A2").Value
  For ct = Par1 To Par2 Step 1.5
       tot = tot + ct
  Next

  Range("A3") = tot
End Sub
```


Istruzioni cicliche

- Calcolare la somma di tutti i numeri interi compresi fra par1 e par2

```
Sub sommaNumeri()
   Dim Par1 As Integer
   Dim Par2 As Integer
   Dim tp As Integer
   Dim tot As Integer
   Dim i As Integer

   Par1 = Range("A1").Value
   Par2 = Range("A2").Value

   If Par1 > Par2 Then
     tp = Par1
     Par1 = Par2
     Par2 = tp
   End If
   tot = 0
   For i = Par1 To Par2
     tot = tot + i
   Next
   Range("A3") = tot
End Sub
```

Istruzione with

- Consente di eseguire una serie di istruzioni contenenti riferimenti ripetuti a un unico oggetto o struttura

```
With oggetto
   istruzioni
End With
```

  - L'argomento di With sono le parti che non cambiano e precedono nella gerarchia le parti che cambiano
    - cells
      - Value
      - Row
      - Column
      - Interior
        - » Colore Cells(x,y).Interior.Color
        - » Motivo Cells(x,y).Interior.Pattern

```
With Cells(x,y).Interior
     .Color =  2
     .Pattern = xlSolid
End With
```


Istruzione With

- Dato il foglio di lavoro nelle Celle da A1 a D7 colorare con
  - sfondo rosso le celle vuote (color 4)
  - Sfondo blu le celle con i numeri (color 6)
  - Sfondo giallo le celle con le date (color 7)

Istruzione With

```
Sub colora()
   Dim r As Range

   For Each r In Range("A1", "D7")
      If IsEmpty(r) Then
        r.Interior.ColorIndex = 4
        r.Interior.Pattern = xlSolid
      ElseIf IsDate(r) Then
        r.Interior.ColorIndex = 7
        r.Interior.Pattern = xlSolid
      ElseIf IsNumeric(r) Then
        r.Interior.ColorIndex = 6
        r.Interior.Pattern = xlSolid
      End If
   Next
End Sub
```

```
Sub colora()
   Dim r As Range

   For Each r In Range("A1", "D7")
   With r
      If IsEmpty(.Value) Then
         With .Interior
           .ColorIndex = 4
           .Pattern = xlSolid
         End With
      ElseIf IsDate(.Value) Then
        With .Interior
           .ColorIndex = 7
           .Pattern = xlSolid
         End With
      ElseIf IsNumeric(r) Then
        With .Interior
           .ColorIndex = 6
           .Pattern = xlSolid
         End With
      End If
   End With
   Next
End Sub
```


Registratore di macro

- Excel fornisce un utile strumento per poter registrare il codice generato in base alle azioni che compiamo mentre lavoriamo.
- In questo modo è possibile esaminare il codice prodotto e quindi regolarsi sui comandi da impiegare
- Per attivare questa funzionalità:

  Strumenti -> Macro -> Registra macro
  - Viene chiesto un nome da dare alla macro
  - Inizia la registrazione che termina schiacciando il tasto STOP

  - Il codice generato si troverà nella cartella moduli di VBA

Registratore di Macro

– Utilizziamo il registratore di macro per registrare i passi relativi alla creazione di un grafico a partire dai dati presenti in una tabella
  - Si ha un insieme di valori nel range A2:B6
  - Si deve creare un grafico di dispersione

Registratore di Macro

```
Sub Grafico()
Charts.Add
    ActiveChart.ChartType = xlXYScatterSmooth
    ActiveChart.SetSourceData Source:=Sheets("Foglio3").Range("A2:B6"), PlotBy _
        :=xlColumns
    ActiveChart.SeriesCollection(1).Name = "=""valori Sperimentali"""
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Foglio3"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "valori Sperimentali"
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "X"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Y"
    End With
    With ActiveChart.Axes(xlCategory)
        .HasMajorGridlines = False
        .HasMinorGridlines = True
    End With
    With ActiveChart.Axes(xlValue)
        .HasMajorGridlines = True
        .HasMinorGridlines = False
    End With
End Sub
```