



## Le Funzioni Condizionali

Una valida alternativa ad una condizione Vera o Falsa può essere la dichiarazione If ... Then ... ElseIf in quanto si comporta come l'espressione If ... Then ... Else, salvo che offre un maggior numero di scelte, se necessario. La sintassi è la seguente:

```
If Condizione1 Then
    Istruzioni1
ElseIf Condizione2 Then
    Istruzioni2
ElseIf CondizioneX Then
    IstruzioniX
End If
```

Quando viene utilizzata questa istruzione, verrà esaminata innanzitutto *Condizione1*, se risulta essere Vera, il programma eseguirà *Istruzioni1* e cessa l'esame delle altre condizioni, mentre se *Condizione1* è falsa, verrà esaminata *Condizione2* e nel caso risulti essere vera verrà eseguita *Istruzioni2*. In sostanza, ogni volta che una condizione è falsa, il programma continuerà l'esame delle condizioni fino a trovarne una che sia vera e una volta che è stata trovata verrà eseguita la sua istruzione e il programma terminerà l'esame condizionale quando incontra l'istruzione End If . Ecco un esempio:

```
Sub Test()
    Dim eta As Byte
    eta = 32
    If eta <= 18 Then
        MsgBox ("Fascia Età : " & "Ragazzo")
    ElseIf eta < 55 Then
        MsgBox ("Fascia Età : " & "Adulto")
    End If
End Sub
```

Ciò produrrebbe:



**Fig. 1**

Esiste però la possibilità che nessuna delle condizioni indicate sia vera, in questo caso, è necessario fornire una condizione "generica" e viene fatta con un ultimo Else, che deve essere l'ultima nella lista di condizioni e deve agire se nessuna delle condizioni precedenti è risultata vera. La formula da usare potrebbe essere:



```
If Condizione1 Then
    Istruzioni1
ElseIf Condizione2 Then
    Istruzioni2
ElseIf CondizioneX Then
    IstruzioniX
Else
    CondizioneG
End If
```

Ecco un esempio:

```
Sub Test()
    Dim eta As Byte
    eta = 65
    If eta <= 18 Then
        MsgBox ("Fascia Età : " & "Ragazzo")
    ElseIf eta < 55 Then
        MsgBox ("Fascia Età : " & "Adulto")
    Else
        MsgBox ("Fascia Età : " & "Anziano")
    End If
End Sub
```

Ciò produrrebbe:



**Fig. 2**

### ■ Dichiarazioni condizionali e funzioni

Come abbiamo già visto, sappiamo che una funzione viene utilizzata per eseguire un compito specifico e produrre un risultato. Ecco un esempio:

```
Private Function etaF()
    Dim eta
    eta = InputBox("Inserisci l'età")
    etaF = ""
End Function
```

Quando una funzione esegue il suo compito, può incontrare diverse situazioni, alcune delle quali avrebbero bisogno di essere controllate per la veridicità o la negazione, ciò significa che le istruzioni condizionali possono aiutare una procedura con la sua assegnazione. Per meglio comprendere, sappiamo che una funzione ha lo scopo di restituire un valore, a volte però deve svolgere alcuni compiti i cui risultati potrebbero portare a risultati diversi.



Una funzione può restituire un solo valore (abbiamo visto che, passando argomenti per riferimento, è possibile effettuare una procedura di restituire più di un valore), ma si può fare il rendering di un risultato a seconda di un particolare comportamento. Se una funzione richiede una risposta da parte dell'utente, in quanto l'utente può fornire risposte diverse, è possibile trattare ogni risultato diverso. Si consideri la seguente funzione:

```
Private Function etaF()  
    Dim eta  
    eta = InputBox("Inserisci l'età")  
    If eta <= 18 Then  
        etaF = "Ragazzo"  
    ElseIf eta < 55 Then  
        etaF = "Adulto"  
    End If  
End Function  
  
Sub Test()  
    Dim tipo  
    tipo = etaF  
    MsgBox ("Fascia di età : " & tipo)  
End Sub
```

A prima vista, questa funzione sembra a posto. L'utente è invitato a fornire un numero. Se l'utente inserisce un numero inferiore a 18 (escluso), la funzione restituisce "Ragazzo". Ecco un esempio del programma:

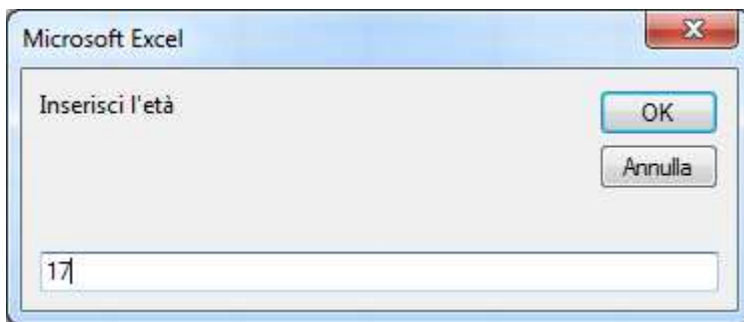


Fig. 3



Fig. 4

Se invece l'utente fornisce un numero compreso tra 18 (incluso) e 55, la funzione restituisce "Adulto". Ecco un altro esempio del programma:

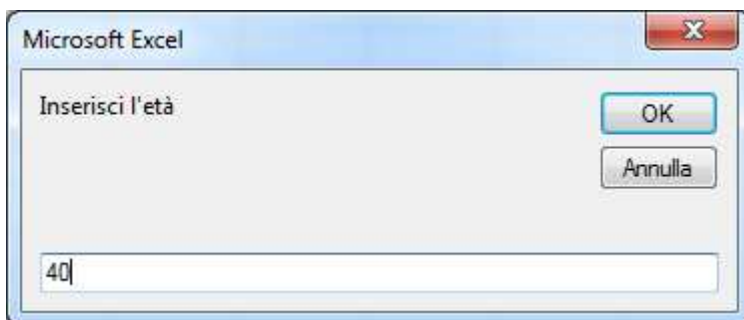


Fig. 5



Fig. 6

Si deve tener presente che i valori restituiti sono dovuti esclusivamente alle istruzioni condizionali inserire e non alla funzione, e che la condizione viene verificata quando è Vera, pertanto se viene inserito un valore che non si "adatta" alle istruzioni la funzione non restituisce nessun valore.



Nel nostro esempio, se l'utente inserisce un numero superiore a 55 (escluso), la funzione non eseguirà nessuna delle dichiarazioni, ciò significa che l'esecuzione raggiungerà la linea End Function senza trovare un valore da restituire. Ecco un altro esempio del programma:

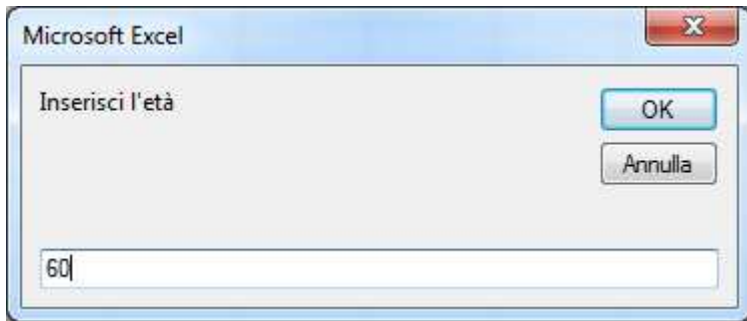


Fig. 7



Fig. 8

Per risolvere questo problema, si dispone di varie alternative, se la funzione utilizza una condizione If ... Then, è possibile creare una condizione Else che abbraccia un valore diverso da quelli validati in precedenza. Ecco un esempio:

```
Private Function etaF()  
    Dim eta  
    eta = InputBox("Inserisci l'età")  
    If eta <= 18 Then  
        etaF = "Ragazzo"  
    ElseIf eta < 55 Then  
        etaF = "Adulto"  
    Else  
        etaF = "Anziano"  
    End If  
End Function  
  
Sub Test()  
    Dim tipo  
    tipo = etaF  
    MsgBox ("Fascia di età : " & tipo)  
End Sub
```

Questa volta, la condizione Else viene eseguita se nessun valore si può applicare alle condizioni If o ElseIf. Ecco un altro esempio:

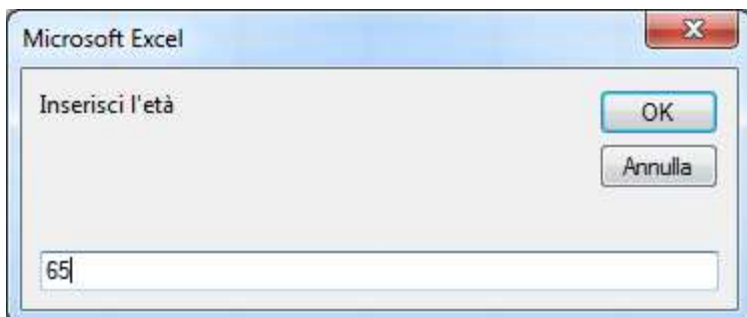


Fig. 9



Fig. 10

Un'altra alternativa è quella di fornire un ultimo valore di ritorno prima della linea End Function, in questo caso, se l'esecuzione raggiunge la fine della funzione, troverebbe dei dati da restituire, ma si dovrebbe conoscere il dato da restituire.



Ciò dovrebbe essere fatto nel modo seguente:

```
Private Function etaF()  
    Dim eta  
    eta = InputBox("Inserisci l'età")  
    If eta <= 18 Then  
        etaF = "Ragazzo"  
    ElseIf eta < 55 Then  
        etaF = "Adulto"  
    End If  
  
    etaF = "Anziano"  
End Function
```

Se la funzione utilizza una condizione If, entrambe le implementazioni produrrebbero lo stesso risultato.

### ■ **Uso della funzione Immediata IIF**

La funzione **IIf** può essere usata anche al posto di uno scenario If ... Then ... ElseIf, infatti quando la funzione viene richiamata, l'espressione viene controllata e, come abbiamo già visto, se l'espressione è vera, la funzione restituisce l'argomento True e ignora l'ultimo argomento. Per utilizzare questa funzione come alternativa alla dichiarazione If ... Then ... ElseIf, se l'espressione è falsa, anziché immediatamente restituire il valore dell'argomento False, è possibile tradurre la parte in una nuova funzione Iif. La sintassi diventerebbe:

```
Public Function IIf( _  
    ByVal Expression As Boolean, _  
    ByVal TruePart As Object, _  
        Public Function IIf( _  
            ByVal Expression As Boolean, _  
            ByVal TruePart As Object, _  
            ByVal FalsePart As Object _  
        ) As Object  
    ) As Object
```

In questo caso, se l'espressione è falsa, la funzione restituisce il valore True e si ferma. Ecco esempio:

```
Sub Test()  
    Dim eta As Byte, fasciaE As String  
    eta = 74  
  
    fasciaE = IIf(eta <= 18, "Ragazzo", IIf(eta < 55, "Adulto", "Anziano"))  
    MsgBox ("Fascia eta : " & fasciaE)  
End Sub
```

Abbiamo visto che in un If ... Then ... ElseIf è possibile aggiungere il numero di condizioni ElseIf come si desidera, nello stesso tempo, è possibile richiamare il numero di funzioni IIF che si ritengono necessarie:



```
Public Function IIf( _  
    ByVal Expression As Boolean, _  
    ByVal [COLOR="rgb(139, 0, 0)"]TruePart [/COLOR]As Object, _  
    Public Function IIf( _  
        ByVal Expression As Boolean, _  
        ByVal [COLOR="rgb(139, 0, 0)"]TruePart [/COLOR]As Object, _  
        [COLOR="rgb(139, 0, 0)"]Public Function IIf[/COLOR]( _  
            ByVal Expression As Boolean, _  
            ByVal [COLOR="rgb(139, 0, 0)"]TruePart [/COLOR]As Object, _  
            [COLOR="rgb(139, 0, 0)"]Public Function IIf[/COLOR]( _  
                ByVal Expression As Boolean, _  
                ByVal TruePart As Object, _  
                ByVal [COLOR="rgb(139, 0, 0)"]FalsePart [/COLOR]As Object _  
            ) As Object  
        ) As Object  
    ) As Object  
) As Object
```

### ■ Scegliere un valore alternativo

Come abbiamo visto finora, la funzione Choose prende una lista di argomenti e per usarla come alternativa al If ... Then ... ElseIf ... ElseIf, è possibile passare i valori che si giudicano necessari per il secondo argomento. L'indice del primo elemento del secondo argomento sarebbe 1, mentre l'indice del secondo elemento del secondo argomento sarebbe 2, e così via. Quando la funzione viene richiamata, si dovrebbe prima ottenere il valore del primo argomento, allora si devono controllare gli indici dei membri disponibili del secondo argomento. Ecco un esempio:

```
Sub Test()  
    Dim statoC As Byte, tipo As String  
    statoC = 3  
    tipo = Choose(statoC, "Tempo Pieno", "Part Time", "Stagionale", "A chiamata")  
  
    MsgBox ("Tipo di Inquadramento : " & tipo)  
End Sub
```

Ciò produrrebbe:



**Fig. 11**

Finora, abbiamo usato solo le stringhe per i valori del secondo argomento della funzione **Choose**, in realtà, i valori del secondo argomento può essere anche nullo, può essere una costante o una stringa.

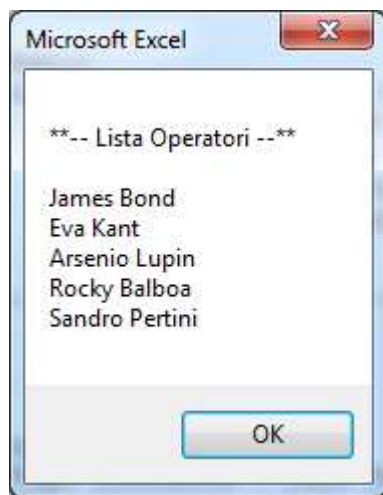




Ecco un esempio:

```
Private Function Moperat()  
    Moperat = "**-- Lista Operatori --**" & vbCrLf & vbCrLf & _  
        "James Bond" & vbCrLf & _  
        "Eva Kant" & vbCrLf & _  
        "Arsenio Lupin" & vbCrLf & _  
        "Rocky Balboa" & vbCrLf & _  
        "Sandro Pertini"  
End Function  
  
Sub Test()  
    Dim tipo As Byte, Result$  
    tipo = 3  
    Result = Choose(tipo, _  
        "Tipo Contratto : Full Time", _  
        "Tipo Contratto : Part Time", _  
        Moperat, _  
        "Tipo Contratto : Stagionale")  
    MsgBox (Result)  
End Sub
```

Ciò produrrebbe:



**Fig. 12**

I valori del secondo argomento possono anche essere di diversi tipi.

### Il passaggio a un valore alternativo

La funzione **Switch** è un metodo alternativo alla condizione If ... Then ... ElseIf ... ElseIf e l'argomento di questa funzione viene passato come un elenco di valori, come visto in precedenza, ogni valore viene passato come una combinazione di due valori:

*CondizioneX, StatementX*

Come si accede alla funzione, il compilatore controlla ogni condizione, se una condizione è vera, la sua dichiarazione viene eseguita, e se una condizione Y è falsa, il compilatore la salta. È anche possibile fornire un maggior numero di queste combinazioni.



Ecco un esempio:

```
Private Enum tipoC
    TPn
    PTm
    Stg
    Ach
End Enum

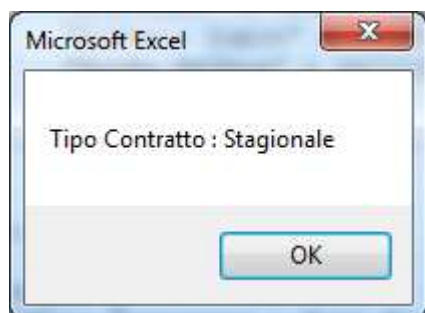
Sub Test()
    Dim tipo As tipoC, Result As String

    tipo = tipoC.Stg
    Result = "Sconosciuto"

    Result = Switch(tipo = tipoC.TPn, "Tempo Pieno", _
        tipo = tipoC.PTm, "Part Time", _
        tipo = tipoC.Stg, "Stagionale", _
        tipo = tipoC.Ach, "A Chiamata")

    MsgBox ("Tipo Contratto : " & Result)
End Sub
```

Ciò produrrebbe:



**Fig. 13**

In una condizione If ... Then ... ElseIf ... ElseIf, abbiamo visto che c'è una possibilità che nessuna delle condizioni si adatterebbe, nel qual caso è possibile aggiungere una ultima dichiarazione Else. La funzione Switch supporta anche questa situazione se si utilizza un numero, un carattere o una stringa e per fornire questa ultima alternativa, invece di un CondizioneX, si deve immettere un valore Vero e includere la dichiarazione. Ecco un esempio:





```
Sub Test()  
  Dim tipo As Byte, Result As String  
  tipo = 12  
  Result = Switch(tipo = 1, "Tempo Pieno", _  
    tipo = 2, "Part Time", _  
    tipo = 3, "A Chiamata", _  
    tipo = 4, "Stagionale", _  
    True, "Sconosciuto")  
  
  MsgBox ("Tipo Contratto : " & Result)  
End Sub
```

Ciò produrrebbe:



**Fig. 14**

Ricorda che puoi utilizzare anche un carattere. Ecco un esempio:

```
Sub Test()  
  Dim sesso As String, Result As String  
  sesso = "H"  
  Result = Switch(sesso = "f", "Femmina", _  
    sesso = "F", "Femmina", _  
    sesso = "m", "Maschio", _  
    sesso = "M", "Maschio", _  
    True, "Sconosciuto")  
  
  MsgBox ("Tipo Sesso: " & Result)  
End Sub
```

Ciò produrrebbe:



**Fig. 15**

