



La Funzione MsgBox e InputBox

Abbiamo usato questa semplice funzione molto spesso in questo corso, adesso vediamola più da vicino e come possiamo personalizzarla per i nostri progetti. La funzione **MsgBox()** ci permette di mostrare a video un box che riporterà un avviso permettendo così all'utente di scegliere l'operazione più idonea da eseguire. Questo comando ci è utile quando stiamo per mandare in esecuzione una determinata procedura e vogliamo ottenere il consenso dall'utente, oppure la possiamo usare nella gestione degli errori, ci può avvisare e impedire l'esecuzione di una routine che porterebbe il programma alla generazione di un errore con conseguente blocco dell'esecuzione del nostro progetto, ma può chiederci una ulteriore conferma per operazioni "delicate" - tipo cancellazione di file - in sostanza è una funzione che ci permette di comunicare da vicino con l'applicazione che stiamo usando.

Finora abbiamo visto marginalmente l'uso di MsgBox, in questa lezione cercheremo di approfondire le sue potenzialità e il suo uso. Non ha una sintassi particolarmente difficile ma il suo uso "avanzato" ci permette di controllare le varie procedure e gestire tutti gli eventi che abbiamo istanziato nel nostro programma, in sostanza è una funzione di VBA abbastanza semplice da usare che riporta a video un messaggio con un'icona e dei pulsanti predefiniti, a cui si risponde premendo su uno di essi. La sintassi è la seguente :

MsgBox(prompt[, buttons] [, title] [,helpfile ,context]) oppure italianizzando il comando
MsgBox(Messaggio[, Pulsanti] [, Titolo] [, Fileaiuto , Contesto]) Dove

- *Prompt o Messaggio* : indica il messaggio che sarà visualizzato nella finestra di dialogo.
- *buttons o Pulsanti* : indica il valore numerico dei pulsanti da visualizzare nella finestra di dialogo.
- *Title o Titolo* : Indica il titolo della finestra di dialogo e va scritto fra virgolette
- *helpfile, FileAiuto e context o Contesto* : sono relativi alla guida dell'applicazione ma non sono indispensabili

Ricordiamoci che non abbiamo nessun controllo della posizione in cui verrà visualizzato il Box sullo schermo, vediamo ora qualche esempio :

```
Sub box1()  
MsgBox "Ciao a tutti"  
End Sub
```

Questo codice ci riporta a video un messaggio come questo

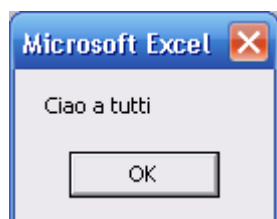


Fig. 1

Come potete vedere abbiamo ommesso alcune espressioni nella sintassi appena esposta, ma il nostro box ci appare con il testo che abbiamo inserito, e con la barra del titolo di default [Microsoft Excel], inoltre è possibile anche far apparire il box usando il valore di *Variabili* oppure possiamo anche usare anche una forma più evoluta e personalizzare il nostro messaggio di avviso come meglio crediamo, vediamone un esempio



```
Sub box2()  
MsgBox "Ciao a tutti", vbCritical + vbOKOnly, "Funzione MsgBox Semplice"  
End Sub
```

Che ci riporta a video un messaggio come questo :



Fig. 2

Notiamo subito che è cambiato il titolo nella barra della finestra ed è comparsa un'icona rossa con una X bianca, possiamo anche cambiare tipo di icona in questo modo :

```
Sub box3()  
MsgBox "Ciao a tutti", vbQuestion + vbOKOnly, "Funzione MsgBox Semplice"  
End Sub
```

E ci verrà riportato a video un messaggio come questo



Fig. 3

Vedendo i codici esposti e i vari box che ci sono apparsi possiamo dire che :

Prompt: è il messaggio che verrà visualizzato, nel nostro caso è *Ciao a tutti*

buttons : è il tipo di pulsante e relativa icona, nel nostro caso è rappresentato dal codice *vbCritical* + *vbOKOnly*

title : è il titolo della finestra, nel nostro caso è *Funzione MsgBox Semplice*

Ora però possiamo anche fare un altro "passo avanti" nell'uso di questa funzione, a volte è difficile ricordare tutti i comandi e potremmo usarla anche in un altro modo. Abbiamo detto poco sopra che *buttons* indica il valore numerico dei pulsanti da visualizzare nella finestra di dialogo, ma finora non abbiamo esposto valori numerici, abbiamo solo rappresentato i pulsanti con delle "parole chiave" tipo : *vbCritical* + *vbOkOnly*, oppure *vbQuestion* + *vbOkOnly* vediamo questo aspetto modificando il codice finora usato in questo modo :



```
Sub box4()  
MsgBox "Ciao a tutti", 0 + 16, "Funzione MsgBox Avanzata" '  
End Sub
```

L'esecuzione di questo codice ci riporta questo avviso



Fig. 4

Come possiamo vedere l'avviso cambia solo nel titolo che abbiamo modificato in *Funzione MsgBox Avanzata* ma il resto del Box è uguale, eppure il codice è rappresentato in maniera diversa, possiamo vedere che non compaiono più le parole chiave usate in precedenza ma abbiamo inserito dei valori numerici separati da virgole e seguiti dal titolo della finestra. A mio avviso usarla in questo modo è molto più semplice da ricordare e avremmo meno codice da scrivere. Esponiamo ora con una tabella come vengono interpretati i valori da VBA, che ci aiuterà nell'interpretazione di quanto finora esposto. L'argomento buttons indica il valore numerico dei pulsanti da visualizzare nella finestra del Box e sono così rappresentati

Valore	Descrizione Pulsanti	Comando	Icona
0	Ok	vbOKOnly	<input type="button" value="Ok"/>
1	Ok - Annulla	vbOKCancel	<input type="button" value="Ok"/> <input type="button" value="Annulla"/>
2	Interrompi - Riprova - Ignora	vbAbortRetryIgnore	<input type="button" value="Interrompi"/> <input type="button" value="Riprova"/> <input type="button" value="Ignora"/>
3	Si - No - Annulla	vbYesNoCancel	<input type="button" value="Si"/> <input type="button" value="No"/> <input type="button" value="Annulla"/>
4	Si - No	vbYesNo	<input type="button" value="Si"/> <input type="button" value="No"/>
5	Riprova - Annulla	vbRetryCancel	<input type="button" value="Riprova"/> <input type="button" value="Annulla"/>

Fig. 5

Sempre a questo argomento possiamo associare un'icona da visualizzare nel Box identificata da un valore numerico seguendo questa tabella:



Valore	Descrizione	Comando	Icona
16	Avviso Grave	vbCritical	
32	Domanda	vbQuestion	
48	Esclamazione	vbExclamation	
64	Informazione	vbInformation	

Fig. 6

Abbiamo detto poco sopra che la funzione MsgBox *restituisce* un valore, questo valore rappresenta *il pulsante che abbiamo premuto*, infatti come facciamo a sapere quale pulsante, e di conseguenza, quale scelta ha fatto l'utente? Ora dobbiamo fare una piccola parentesi, *nelle lezioni precedenti abbiamo sempre esposto MsgBox come un semplice avviso*, invece possiamo usarla anche quando dobbiamo prendere delle decisioni, in pratica si sta dimostrando l'estrema versatilità di questa funzione, unica cosa è fondamentale sapere quale tasto è stato premuto dall'utente per consentire o negare l'esecuzione di una procedura.

A questo punto non diventa solo una semplice funzione che rimanda un avviso, ma prende campo un aspetto più importante, cioè può permettere l'esecuzione di una procedura oppure indicarci quale operazione stiamo per eseguire e richiedere un'ulteriore conferma. Vediamo ora con una tabella quali sono i valori che vengono restituiti da MsgBox e poi con qualche riga di codice ne vediamo il suo uso all'interno di una procedura

Valore	Pulsante Selezionato	Comando
1	Ok	vbOk
2	Cancel	vbCancel
3	Abort	vbAbort
4	Retry	vbRetryCancel
5	Ignore	vbIgnore
6	Yes	vbYes
7	No	vbNo

Fig. 7

In base alla tabella sopra esposta possiamo dire che se l'utente preme il tasto Ok il valore restituito sarà 1, mentre se preme il pulsante Yes il valore restituito sarà 6, di conseguenza possiamo usare la funzione MsgBox anche in altri contesti diversi dal semplice avviso, ma usarla anche per operare delle scelte ed usando le Variabili – come abbiamo detto all'inizio – per ottimizzare sia la funzione ma soprattutto l'uso che ne viene fatto. Vediamo qualche esempio di codice



```
Sub prova()  
Dim Risp As Integer  
Risp = MsgBox("Prova funzione MsgBox", 1 + 64, "Funzione MsgBox Avanzata")  
If Risp = 1 Then  
    MsgBox "Hai schiacciato il pulsante Ok", 1+48, "Funzione MsgBox Avanzata"  
Else  
    Exit Sub  
End If  
End Sub
```

Con il codice sopra esposto ci compare una finestra come questa



Fig. 8

E cliccando sul pulsante Ok ci comparirà un messaggio del genere

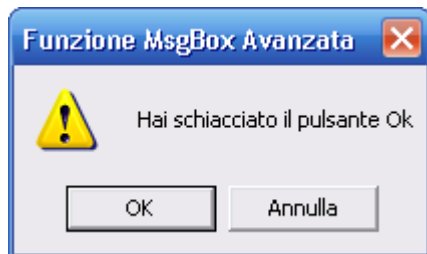


Fig. 9

Credo che sia abbastanza eloquente come solo sostituendo le condizioni da verificare e le procedure da eseguire possiamo utilizzare questa funzione con scopi ben diversi dal solo avviso, ma possiamo integrarla con le scelte che andiamo ad operare nel proseguo del nostro programma, costituendo così un'ossatura stabile e logica del nostro codice sia per quanto riguarda la gestione degli errori che richiedendo conferma all'utente di quanto si appresta a fare e in base alle scelte che effettua indirizzare il flusso del programma nella direzione appropriata



■ La Funzione InputBox

Se vogliamo che l'utente possa operare delle scelte su come usare la procedura possiamo usare la funzione MsgBox che abbiamo già visto nella lezione precedente oppure la funzione InputBox, la quale ci permette di ottenere un input dall'utente, la sintassi generale è la seguente:

InputBox (Messaggio) [, Titolo, Default, XPos, YPos, File Aiuto, Contesto]

Messaggio è una stringa usata per indicare all'utente quale informazione deve inserire, ed è l'unico argomento richiesto, tutti gli altri sono opzionali.

Titolo è una stringa usata come titolo per la finestra di dialogo

Default è una stringa per fornire un valore di Default per l'input dell'utente

XPos e *YPos* sono espressioni numeriche che forniscono le coordinate dove deve apparire la finestra di dialogo, *XPos* è la distanza orizzontale dal lato sinistro della finestra e *YPos* è la distanza verticale dal lato superiore della finestra, sono argomenti opzionali, ma fate attenzione se li usate perchè se specificate delle posizioni troppo grandi per questi argomenti si corre il rischio di non far apparire la finestra sullo schermo

FileAiuto è una stringa che contiene il nome di un file della guida di Windows e *Contesto* è un'espressione numerica che specifica l'argomento nel file della guida relativo alla finestra di dialogo che state visualizzando. *FileAiuto* e *Contesto* sono opzionali, ma se specificate *FileAiuto* dovete specificare anche *Contesto*, e quando specificate un file della guida per una finestra di dialogo di input, VBA aggiunge automaticamente un pulsante della Guida (?) alla finestra di dialogo. Vediamo ora un esempio

```
Sub funzione_input()  
Prova_input = InputBox(prompt:="Inserisci il nome di un file: ", Title:="Crea un nuovo  
file", Default:="Newfile")  
End Sub
```

ed otteniamo un finestra dei questo tipo

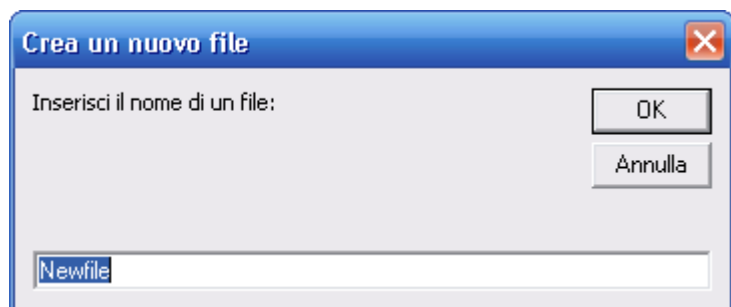


Fig. 10

credo che sia abbastanza semplice ed intuitivo il listato esposto, infatti vediamo che quando digitato nel codice appare nella finestra di dialogo, ora possiamo passare all'argomento appena accennato all'inizio cioè prendere delle decisioni, naturalmente le nostre procedure non possono veramente "Prendere delle decisioni" allo stesso modo di un essere umano, ma bensì possono scegliere tra diversi percorsi di azioni predefinite, basandosi su semplici condizioni e prendendo delle decisioni al solo verificarsi di determinati eventi. Possiamo dire che usiamo le istruzioni di scelta di VBA, definite in una condizione oppure in un insieme di condizioni per cui VBA esegue un blocco di codice della nostra procedura oppure un altro blocco di codice.