



Scrittura di nuove macro e procedure

Se esaminiamo una macro ottenuta con il *Registratore di macro* si potrà notare che hanno tutte alcune caratteristiche in comune. Per esempio registriamo due macro, una che formatti una selezione di celle (A1:D10) e applichi al carattere il formato grassetto e l'altra che applichi il colore rosso allo stesso intervallo di celle. Operando dal Registratore di macro, come abbiamo già visto, per la prima macro, quella che formatta le celle in grassetto otteniamo il seguente listato.

```
1) Sub Grassetto()  
2) '  
3) ' Macro Grassetto  
4) ' Macro registrata il .....  
5) '  
6) Range("A1:D10").Select  
7) Selection.Font.Bold = True  
8) Range("A1").Select  
9) End Sub
```

E per la seconda macro che applica il colore rosso al carattere otteniamo il seguente listato

```
1) Sub Colore()  
2) '  
3) ' colore Macro  
4) ' Macro registrata il .....  
5) '  
6) Range("A1:D10").Select  
7) Selection.Font.ColorIndex = 3  
8) Range("A1").Select  
9) End Sub
```



*Il codice sorgente di una macro in un modulo **NON** include un numero davanti ad ogni linea, La numerazione delle linee dei listati rappresentati è stata usata solo per facilitare l'identificazione e la discussione di particolari linee nei vari listati*

Come abbiamo esordito all'inizio affermando che tutte le macro hanno caratteristiche in comune, con questi due listati possiamo iniziare a notare qualche affinità. La *riga 1* è sempre l'inizio della macro e ogni macro VBA inizia con il termine **Sub** seguito dal nome della macro. La linea che contiene la parola chiave Sub e il nome della macro viene detta "*dichiarazione della macro*", in quanto è quella che rende noto al VBA l'esistenza della macro stessa. Il nome della macro, a sua volta, è sempre seguito da una coppia di *parentesi vuote*(), il cui scopo e utilizzo verrà affrontato più avanti nel corso, per il momento basta solo sapere che le due parentesi devono sempre comparire dopo il nome della macro nel codice sorgente. Un'altra affinità è rappresentata dalla parola chiave **End Sub** alla *riga 9* che segnala al VBA che è stata raggiunta la fine della macro.

Dalla *riga 2* alla *riga 5* dei due listati sono presenti dei commenti. Un commento è una riga della macro che non contiene istruzioni ma che fanno parte della macro stessa, generalmente viene usato per dei chiarimenti sull'azione che sta per compiere la macro stessa. Come si può notare, ogni riga di commento inizia con un apostrofo (') e il VBA tratta ogni riga che comincia con un apostrofo come un commento e non processa il testo al suo interno. Subito dopo la dichiarazione *Sub nome_della_macro ()* segue il corpo della macro che può includere varie righe di codice e/o di commento.



Nel caso della macro *Grassetto* le righe da 6 a 8 costituiscono il corpo della macro, così come la macro *Colore*. Ogni riga del corpo della macro consiste in uno o più enunciati che rappresentano una serie di parole chiave e altri simboli che assieme costituiscono un'istruzione VBA completa.

Gli enunciati VBA di una macro registrata contengono le istruzioni che svolgono azioni equivalenti a quelle che sono state eseguite quando si è registrata la macro stessa. Per esempio la riga 6 della macro *Grassetto* seleziona il Range di celle A1:D10 così come la stessa riga della macro *Colore*, mentre invece la riga 7 della macro *Grassetto* fissa il carattere al formato grassetto, la stessa riga della macro *Colore* fissa il colore del testo a rosso. La riga 8 dei due listati seleziona la cella A1 in quanto dopo aver formattato le celle abbiamo cliccato sulla cella A1.

Quando si esegue una macro il VBA parte dalla prima riga del corpo della macro (la prima dopo la dichiarazione) eseguendo le istruzioni contenute in essa in sequenza, da sinistra verso destra, poi passa alla riga successiva ed esegue le istruzioni in essa contenute e così di seguito fino a raggiungere la fine della macro, definita dall'enunciato *End Sub*. Possiamo inoltre notare che alcune parti del testo della macro sono visualizzate in colori differenti, i commenti sono visualizzati in colore verde, mentre le parole chiave *Sub* e *End Sub* e altre del VBA sono di colore blu, mentre il testo rimanente della macro è di colore nero, a indicare che si tratta di dati e enunciati creati dall'utente. Il colore con cui VBA contraddistingue le varie parti di testo sullo schermo servono a rendere più chiara la parte della macro o di un enunciato che si sta esaminando.

■ Modifica del testo di una macro

Per apportare delle modifiche al codice sorgente di una macro in un modulo si usano comandi e tecniche che risultano già famigliari a chi usa Windows, la modifica del testo in un modulo visualizzato nella finestra Codice avviene esattamente come quando si edita un testo in Word o in qualsiasi altro elaboratore di testi, per aggiungere, eliminare, selezionare, tagliare etc. si useranno quindi i medesimi comandi da tastiera o col mouse. Le modifiche apportate a un modulo si possono salvare tramite il comando **File - Salva** dell'Editor di VB oppure facendo clic sul pulsante *Salva* della barra degli strumenti. Ogni modifica effettuata su un modulo viene comunque salvata quando si salva la cartella di lavoro che contiene il modulo stesso.

Per fare un esempio di come effettuare modifiche a una macro prendiamo in esame la prima macro registrata (*Grassetto*) e andiamo a modificarla, oltre ad applicare il formato grassetto vogliamo anche cambiare dimensioni al carattere. Possiamo procedere in questo modo, portiamoci nell'Editor di VBA nella macro *Grassetto* e spostiamo il punto di inserimento (il cursore) alla fine della riga 4 e premiamo Invio per inserire una riga vuota. Digitiamo un apostrofo (') e digitiamo poi il testo del commento (per esempio: la linea successiva cambia la dimensione del carattere) a questo punto nella riga successiva al commento dobbiamo solo scrivere il codice VBA che esegua l'operazione richiesta.

Possiamo aiutarci col registratore di macro, quando abbiamo trattato il registratore abbiamo detto che una volta attivato **registra tutte le operazioni svolte dall'operatore** e le converte in codice VBA, pertanto possiamo ritornare a Excel, attivare la registrazione di una nuova macro, selezionare l'intervallo di celle riportato nella macro *Grassetto* e modificare la dimensione del carattere. Una volta eseguita questa operazione fermiamo la registrazione e torniamo nell'Editor VBA per vedere il codice, che si presenta in questo modo:



```
Sub Macro2()  
'  
' Macro2 Macro  
' Macro registrata il .....  
'  
    Range("A1:D10").Select  
    With Selection.Font  
        .Name = "Arial"  
        .Size = 12  
        .Strikethrough = False  
        .Superscript = False  
        .Subscript = False  
        .OutlineFont = False  
        .Shadow = False  
        .Underline = xlUnderlineStyleNone  
        .ColorIndex = xlAutomatic  
    End With  
End Sub
```

Possiamo adesso prelevare il codice che ci interessa e aggiungerlo alla macro *Grassetto* in modo che compia tutte e due le operazioni, cioè che applichi il formato grassetto e inserisca il carattere Arial da 12 Pt. Possiamo ridurre il listato riportato a queste istruzioni

```
With Selection.Font  
    .Name = "Arial"  
    .Size = 12  
End With
```

Che sono quelle che si occupano di modificare la dimensione del carattere, vedremo più avanti il significato delle altre istruzioni quando tratteremo il *Ciclo With*, per il momento consideriamo la sola modifica di una macro utilizzando il registratore di macro quando non si conosce il codice VBA da inserire. A questo punto possiamo prelevare il codice riportato e aggiungerlo alla macro *Grassetto* semplicemente selezionandolo e copiandolo premendo consecutivamente i tasti **Ctrl+C** oppure passando dal menu **Modifica - Copia**.

A questo punto andiamo nella macro *Grassetto* e posizioniamo il cursore sulla riga sotto al commento che abbiamo inserito in cui abbiamo scritto che veniva modificata la dimensione del carattere e premiamo consecutivamente i tasti **Ctrl+V** oppure dal menu **Modifica - Incolla**. Adesso la macro è diventata

```
Sub Grassetto()  
'  
' Grassetto Macro  
' Macro registrata il .....  
' La linea successiva cambia la dimensione del carattere  
Range("A1:D10").Select  
    With Selection.Font  
        .Name = "Arial"  
        .Size = 12  
    End With  
Selection.Font.Bold = False  
Range("A1").Select  
End Sub
```



A questo punto la macro **Grassetto** è stata modificata usando il registratore di macro per trasformare le nostre azioni compiute sul foglio in codice sorgente e se eseguiamo la macro adesso oltre ad applicare il formato grassetto applica anche il carattere Arial da 12 Pt. Si consiglia di aggiungere commenti al corpo della macro che spieghino i motivi e lo scopo delle modifiche apportate a una macro che potranno facilitare l'identificazione delle parti revisionate e far capire meglio che cosa fanno le modifiche

■ Scrittura di nuove macro e procedure

Per scrivere una propria macro senza ricorrere al registratore di macro si può digitare la macro entro un modulo esistente o creare un nuovo modulo, finora abbiamo utilizzato il termine macro per identificare sia quelle scritte usando il registratore di macro che quelle "*autonome*" cioè scritte di proprio pugno, esiste però un termine che aiuta a distinguere tra macro registrate e "autonome" e, strettamente parlando, il termine macro dovrebbe essere applicato solo alle macro registrate, mentre invece le macro scritte partendo da zero sono chiamate *procedure sub* (subroutine) o semplicemente *procedure*, pertanto nel proseguo di questo corso verrà utilizzato il termine macro per identificare il codice sorgente ottenuto usando il registratore di macro e procedure per riferirsi al codice VBA scritto dall'utente

Per scrivere una nuova procedura entro un modulo esistente bisogna prima aprire la finestra Codice facendo doppio clic sul modulo in Gestione progetti oppure selezionando il modulo e poi scegliere **Visualizza - Codice**. Per scrivere il testo del codice di una procedura, che venga aggiunta ad un modulo nuovo o a uno esistente, si deve portare il cursore di testo nella finestra Codice sul punto del modulo in cui si vuole digitare la nuova procedura. Si può battere il testo di una nuova procedura in una posizione qualsiasi di un modulo, purchè essa venga comunque posizionata dopo l'enunciato End Sub che segna la fine della procedura precedente e prima dell'enunciato Sub della procedura immediatamente successiva, personalmente consiglio di aggiungere una nuova procedura sempre alla fine del modulo.

Quando si scrive una procedura bisogna specificarne il nome e includere la parola chiave *Sub* e *End Sub* rispettivamente all'inizio e alla fine della procedura stessa, se si trascura uno di questi tre particolari la sintassi della procedura non risulterà corretta e l'Editor di VB visualizzerà un messaggio di errore quando si eseguirà la procedura. Si deve tenere presente che se digitiamo il nome della procedura preceduto dalla parola chiave Sub e battiamo il tasto invio l'Editor completerà la procedura aggiungendo le due parentesi vuote e la parola chiave End Sub

Il primo classico programma che si scrive in un qualsiasi linguaggio di programmazione, per tradizione, è quello che visualizza a video la frase: *Ciao Mondo!* Il listato che segue mostra un programma VBA con una singola procedura che adempie allo scopo.

```
Sub Mondo()  
MsgBox "Ciao Mondo!"  
End Sub
```

Per realizzare questo listato ecco come procedere.

- Aprite una cartella di lavoro di Excel o create una nuova cartella
- Premete i tasti **Alt+F11** per entrare nell'Editor VB
- In *Gestione progetti* selezionate il documento o la cartella in cui si vuole memorizzare il programma
- Scegliere dal menu **Inserisci - Modulo** per aggiungere un nuovo modulo al progetto. L'Editor VB aggiunge un nuovo modulo e apre la finestra Codice
- Cambiate il nome del modulo, risulta più intuitivo riconoscere il contenuto del modulo dal nome, per esempio mettete saluto
- Essendo nuovo il modulo iniziate a scrivere dalla prima riga, in caso contrario accertatevi di scrivere il testo all'inizio di una riga e digitate il listato che abbiamo visto poco sopra premendo invio al termine di ogni riga



L'Editor VB include varie funzioni che aiutano a scrivere le procedure, in primo luogo non appena si preme invio dopo aver digitato la parola chiave Sub seguita dal nome della procedura, automaticamente viene pure aggiunta la riga con la parola chiave End Sub, in tal modo non occorre preoccuparsi di una possibile dimenticanza di questo elemento critico della procedura. Inoltre l'Editor VB comprende una funzionalità detta *informazioni rapide automatiche* e non appena si digita la parola chiave *MsgBox* e si preme la *barra spazio* compare una finestrella rettangolare che mostra l'elenco completo di tutti gli argomenti della procedura o funzione VBA integrata in questione, in questo caso gli argomenti di MsgBox

```
msgbox |  
MsgBox(Prompt, [Buttons As VbMsgBoxStyle = vbOKOnly], [Title], [HelpFile], [Context]) As VbMsgBoxResult
```

Fig. 1

L'argomento che si prevede sia il prossimo ad essere digitato compare in grassetto in questa finestrella (un argomento è un'informazione richiesta dalla procedura per poter svolgere il suo lavoro) la finestrella Informazioni rapide automatiche si chiude non appena si preme Invio per iniziare una nuova riga nella finestra del codice oppure ci si sposta dalla linea del codice con i tasti freccia o col cursore del mouse. Se la finestrella delle informazioni rapide automatiche anziché essere utile dà fastidio, la si può disabilitare (e riattivare) col comando **Strumenti - Opzioni dell'Editor VB**. Una volta immesso il codice sorgente della procedura Mondo la si può eseguire in questo modo:

- Selezionare il comando **Strumenti - Macro** per aprire la finestra di dialogo delle macro
- Selezionare la procedura *Mondo* dall'elenco dei nomi delle macro
- Fare clic sul pulsante Esegui

Quando il VBA esegue la procedura visualizza la finestra di dialogo sotto riportata



Fig. 2

Si consideri l'uso della funzione *MsgBox* a puro titolo di esempio, tratteremo questa interessante funzione molto dettagliatamente nel proseguo del corso, lo scopo della procedura Mondo era solo per mostrare i passaggi da eseguire per scrivere una procedura manualmente