

Capitolo 38: Introduzione a Visual Basic, Applications Edition

1. Introduzione alle Macro VBA in Excel

a. Introduzione alle Macro VBA

Le macro VBA (Visual Basic for Applications) sono sequenze di comandi e funzioni scritte in VBA che automatizzano attività ripetitive in Excel. Permettono di eseguire operazioni complesse in pochi clic, risparmiando tempo e riducendo errori.

b. Visualizzare la scheda Sviluppo

Per accedere agli strumenti VBA è necessario abilitare la scheda 'Sviluppo':
File → Opzioni → Personalizzazione barra multifunzione → Attiva 'Sviluppo'.

c. Conoscere la sicurezza delle macro

Le macro possono contenere codice potenzialmente dannoso, quindi Excel imposta diversi livelli di sicurezza:

- Disattiva tutte le macro senza notifica
- Disattiva tutte le macro con notifica
- Disattiva tutte le macro eccetto quelle firmate digitalmente
- Abilita tutte le macro (non consigliato)

Percorso: File → Opzioni → Centro protezione → Impostazioni Centro protezione → Impostazioni macro.

d. Salvare cartelle di lavoro con macro

Quando si salva un file con macro, è necessario usare il formato .xlsm (Cartella di lavoro con attivazione macro Excel).

Se si salva nel formato .xlsx, le macro verranno eliminate.

Percorso: File → Salva con nome → Tipo file → Cartella di lavoro con attivazione macro (*.xlsm).

Esempi pratici

Registra una macro che colora le celle selezionate

Utilizza 'Registra macro', colora alcune celle, poi interrompi la registrazione.

Visualizza la scheda Sviluppo

Attiva tramite Opzioni Excel la scheda per accedere all'Editor VBA.

Apri il Centro protezione e imposta il livello macro su 'Con notifica'

Controlla che le macro non vengano eseguite automaticamente.

Scrivi una macro che inserisce la data corrente

```
Sub InserisciData()  
ActiveCell.Value = Date  
End Sub
```

Salva il file come .xlsm dopo aver creato una macro

File → Salva con nome → .xlsm per mantenere le macro attive.

Esercizi

Crea una macro che applica il grassetto al contenuto selezionato

Suggerimento: usa 'Selection.Font.Bold = True'.

Verifica se la tua versione di Excel mostra la scheda Sviluppo

Se assente, attivala dalle opzioni della barra multifunzione.

Imposta il livello di sicurezza macro su 'solo firmate digitalmente'

Esplora le implicazioni di sicurezza e la gestione dei certificati.

Scrivi una macro che cancella l'intervallo A1:A10

Utilizza 'Range("A1:A10").ClearContents'.

Salva un file contenente macro e verifica che l'estensione sia corretta

Assicurati che il formato sia .xlsm e non .xlsx.

2. Tipi di Macro VBA e Creazione di Macro in Excel

a. Due tipi di macro VBA

Le macro in VBA si suddividono in due categorie principali:

- Subroutine (Sub): eseguono una serie di istruzioni ma non restituiscono valori.
- Funzioni (Function): restituiscono un valore e possono essere utilizzate direttamente nei fogli di lavoro.

b. Subroutine VBA (Subprocedures)

Le subroutine (Sub) sono blocchi di codice che eseguono compiti ma non restituiscono valori.

Sintassi:

```
Sub NomeSub()  
    ' codice VBA  
End Sub
```

Esempio:

```
Sub SalutaUtente()  
    MsgBox "Ciao, utente!"  
End Sub
```

c. Funzioni VBA (Functions)

Le funzioni VBA restituiscono un valore e possono essere usate nei fogli di Excel come qualsiasi funzione nativa.

Sintassi:

```
Function NomeFunzione(Parametri) As Tipo  
    ' codice  
    NomeFunzione = Valore  
End Function
```

Esempio:

```
Function Quadrato(x As Double) As Double  
    Quadrato = x * x  
End Function
```

d. Creare macro VBA

Puoi creare macro VBA:

- Registrandole con il registratore di macro
- Scrivendole direttamente nell'Editor VBA (Alt+F11)

Per creare manualmente:

1. Apri l'Editor VBA con Alt+F11
2. Inserisci un modulo: Inserisci → Modulo
3. Scrivi il codice
4. Salva il file in formato .xlsm per mantenere le macro attive

Esempi pratici

Sub che evidenzia celle

```
Sub EvidenziaCelle()  
    Selection.Interior.Color = RGB(255, 255, 0)  
End Sub
```

Funzione che calcola il cubo

```
Function Cubo(x As Double) As Double  
    Cubo = x ^ 3  
End Function
```

Sub che inserisce il nome dell'utente in una cella

```
Sub InserisciNome()  
    Range("A1").Value = Environ("Username")  
End Sub
```

Funzione che restituisce TRUE se un numero è pari

```
Function ÈPari(n As Integer) As Boolean  
    ÈPari = (n Mod 2 = 0)  
End Function
```

Crea una Sub per svuotare l'intervallo B2:B10

```
Sub SvuotaIntervallo()  
Range("B2:B10").ClearContents  
End Sub
```

Esercizi

Crea una Sub che colora in rosso tutte le celle selezionate

Usa l'oggetto Selection e la proprietà Interior.Color.

Scrivi una funzione che calcola la media di tre numeri

Parametri: a, b, c. Restituisci $(a+b+c)/3$.

Crea una Sub che mostra l'ora corrente in una MsgBox

Utilizza la funzione VBA Time.

Scrivi una funzione che conta quante vocali ci sono in una stringa

Analizza ogni carattere della stringa.

Registra una macro che ordina una tabella di dati per la prima colonna

Poi analizza il codice generato nell'editor VBA.

3. Registrazione di Macro in VBA

a. Registrare le azioni per creare codice VBA: le basi

Excel permette di registrare una sequenza di azioni e convertirle automaticamente in codice VBA. Questo è utile per automatizzare attività ripetitive senza conoscere la sintassi di programmazione.

b. Registrare una macro: un esempio semplice

Per registrare una macro:

1. Vai alla scheda Sviluppo
2. Clicca su 'Registra macro'
3. Dai un nome alla macro e scegli dove salvarla
4. Esegui le azioni che vuoi automatizzare
5. Clicca su 'Interrompi registrazione'

c. Esaminare la macro

Dopo aver registrato la macro, puoi visualizzare il codice generato:
Alt+F11 → Moduli → seleziona il modulo con la macro registrata.

d. Testare la macro

Per testare la macro:

1. Premi Alt+F8
2. Seleziona la macro e clicca su 'Esegui'
3. Verifica che le azioni vengano ripetute come previsto

e. Modificare la macro

Puoi modificare il codice VBA generato automaticamente per ottimizzare le istruzioni, aggiungere controlli condizionali, o generalizzare il comportamento.

f. Registrazione relativa vs assoluta

La registrazione relativa applica le azioni rispetto alla cella attiva, mentre quella assoluta usa riferimenti fissi. Puoi attivare la modalità relativa dalla scheda Sviluppo prima di registrare.

g. Un altro esempio

Registra una macro che applica un bordo a una tabella e imposta un colore di sfondo. Analizza il codice generato per personalizzarlo ulteriormente.

h. Eseguire la macro

Oltre a premere Alt+F8, puoi:

- Assegnare la macro a un pulsante
- Creare un collegamento rapido
- Usare l'Editor VBA per eseguirla manualmente

i. Riregistrare la macro

Se vuoi aggiornare una macro registrata, puoi semplicemente registrarla di nuovo con lo stesso nome, sovrascrivendo il codice precedente.

Esempi pratici

Registra una macro che colora di giallo l'intervallo A1:A10

Usa la funzione di registrazione e poi esamina il codice prodotto.

Registra una macro con modalità relativa

Sposta la selezione e applica una formattazione. Analizza come cambia il codice.

Modifica una macro registrata per includere un messaggio

Usa MsgBox all'inizio della Sub per visualizzare un messaggio.

Crea un pulsante nella scheda Inserisci e assegnagli una macro

Usa un modulo Form Control per richiamare la macro con un clic.

Registra e rivedi una macro che copia e incolla un intervallo

Prova a modificare gli indirizzi nel codice per renderli dinamici.

Esercizi

Registra una macro che applica il grassetto e l'allineamento centrale

Usa la barra multifunzione e salva la macro in un modulo personale.

Rendi la macro compatibile con più fogli

Sostituisci i riferimenti specifici con ActiveSheet o Cells.

Riregistra una macro esistente per includere più formattazioni

Espandi la macro aggiungendo colore al testo o ai bordi.

Scrivi a mano una macro ispirata a una registrata

Analizza la logica del codice e riscrivilo in modo più efficiente.

Assegna la macro a una combinazione di tasti

Usa Alt+F8 → Opzioni per definire la scorciatoia.

4. Approfondimento sulla registrazione delle macro VBA

a. Salvare le macro nella cartella Macro personale (Personal Macro Workbook)

La cartella Macro personale è un file nascosto che si apre ogni volta che avvii Excel. Salvare macro in questa cartella permette di renderle disponibili in qualsiasi file Excel. Durante la registrazione, scegli 'Cartella macro personale' come destinazione.

b. Assegnare una macro a una combinazione di tasti

Puoi associare una macro a una scorciatoia da tastiera durante la registrazione oppure modificando in seguito:

Alt + F8 → Seleziona macro → Opzioni → Inserisci il tasto di scelta rapida.

c. Assegnare una macro a un pulsante

Vai alla scheda 'Sviluppo' → Inserisci → Pulsante (controllo modulo). Disegna il pulsante nel foglio e scegli la macro da associare. Puoi anche rinominare il pulsante cliccando con il tasto destro.

d. Aggiungere una macro alla barra di accesso rapido

1. Clic su freccia giù in alto a destra → Altri comandi
2. Seleziona 'Macro' dal menu a discesa
3. Aggiungi la macro desiderata
4. Puoi anche cambiare l'icona e il nome visualizzato

Esempi pratici

Registra una macro nella cartella macro personale

Imposta la destinazione della registrazione su 'Cartella macro personale' e crea una macro semplice.

Assegna una scorciatoia da tastiera a una macro

Durante la registrazione, imposta Ctrl+Shift+M come combinazione rapida.

Crea un pulsante che esegue una macro

Usa 'Inserisci' nella scheda Sviluppo per aggiungere il controllo modulo.

Aggiungi una macro alla barra di accesso rapido

Personalizza la barra e inserisci un'icona personalizzata per la macro.

Rinomina un pulsante e cambia il suo stile

Clic destro sul pulsante, scegli 'Modifica testo', cambia il colore del bordo.

Esercizi

Registra una macro che inserisce il tuo nome nella cella A1

Salvala nella cartella macro personale per averla sempre disponibile.

Crea una macro e assegna Ctrl+Shift+T

La macro deve evidenziare l'intervallo selezionato.

Disegna un pulsante che lancia una macro di salvataggio

La macro deve salvare il file corrente con una conferma tramite MsgBox.

Aggiungi una macro di stampa alla barra di accesso rapido

Configura l'icona e prova a usarla in diversi file Excel.

Crea due macro simili e confronta l'utilizzo con e senza pulsante

Valuta il risparmio di tempo nell'utilizzo tramite pulsante.

5. Scrivere codice VBA in Excel

a. Le basi: inserire e modificare codice

Il codice VBA si scrive nell'Editor VBA (Alt + F11). Puoi creare un modulo tramite Inserisci → Modulo e iniziare a scrivere Sub o Function.

b. Il modello a oggetti di Excel (Excel Object Model)

Excel è strutturato come una gerarchia di oggetti (Application > Workbook > Worksheet > Range). Ogni elemento può essere controllato via codice.

c. Oggetti e Collezioni

Un oggetto rappresenta un'entità come un foglio o una cella. Una collezione è un insieme di oggetti simili, es. Workbooks o Worksheets.

d. Proprietà

Le proprietà rappresentano caratteristiche di un oggetto. Ad esempio: Range("A1").Value restituisce o imposta il contenuto di A1.

e. Metodi

I metodi sono azioni che un oggetto può eseguire. Esempio: Range("A1").ClearContents cancella il contenuto di A1.

f. L'oggetto Range

Range è l'oggetto più utilizzato. Può riferirsi a una o più celle. Permette operazioni su valori, formati, formule, ecc.

g. Variabili

Le variabili memorizzano valori temporanei. Sintassi: Dim nome As Tipo. Esempio: Dim somma As Double.

h. Controllo del flusso di esecuzione

Puoi usare If, For, Do While, Select Case per controllare il flusso del codice e la logica di esecuzione.

i. Una macro che non può essere registrata

Alcune macro avanzate (es. uso di cicli, condizioni, manipolazione dinamica) non possono essere registrate e devono essere scritte a mano.

j. Esempi pratici

Usa un oggetto Range per impostare un valore

```
Range("B2").Value = 123
```

Dichiara e usa una variabile

```
Dim nome As String
```

```
nome = "Mario"
```

```
MsgBox nome
```

Ciclo For per scrivere numeri da 1 a 10 in colonna A

```
For i = 1 To 10
```

```
Cells(i, 1).Value = i
```

```
Next i
```

Usa un metodo per cancellare un intervallo

```
Range("A1:A10").ClearContents
```

Usa un If per controllare un valore

```
If Range("A1").Value > 100 Then
```

```
Range("A2").Value = "Alto"
```

```
End If
```

k. Esercizi

Scrivi una Sub che somma i valori da A1 a A10

Usa un ciclo e una variabile accumulatrice.

Crea una funzione che calcola la media di tre numeri

Usa parametri e restituisci la media.

Scrivi un ciclo che evidenzia in giallo le celle non vuote in B1:B20

Usa If e Interior.Color.

Scrivi una macro che controlla se A1 è vuota e inserisce un messaggio

If Range("A1").Value = "" Then ...

Crea una Sub che cambia il colore del testo in base al contenuto

Esempio: rosso per valori < 0.