

Capitolo 39: Creazione di funzioni personalizzate del foglio di lavoro

1. Introduzione alle Funzioni VBA

a. Cos'è una funzione VBA

Le funzioni VBA (User Defined Functions - UDF) permettono di estendere le funzionalità di Excel creando logiche personalizzate. A differenza delle Sub, le funzioni restituiscono un valore e possono essere utilizzate direttamente nei fogli di lavoro di Excel.

b. Sintassi

Sintassi generale:

```
Function NomeFunzione(Parametri) As Tipo
    ' istruzioni
    NomeFunzione = risultato
End Function
```

c. Modalità di attivazione

Una funzione VBA può essere attivata:

- Direttamente da una cella, se inserita in un modulo standard
- Scrivendo il nome della funzione come qualsiasi altra funzione Excel
- Via codice all'interno di un'altra Sub o Function

Attivazione da interfaccia:

- Ribbon: Alt + F11 → Inserisci → Modulo → Scrivi la funzione
- Uso in cella: =NomeFunzione(...)

Esempi pratici

Funzione che restituisce il quadrato di un numero

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

Funzione che verifica se un numero è pari

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

Funzione che concatena nome e cognome

```
Function NomeCompleto(nome As String, cognome As String) As String
    NomeCompleto = nome & " " & cognome
End Function
```

Funzione che calcola l'IVA su un importo

```
Function CalcolaIVA(importo As Double) As Double
    CalcolaIVA = importo * 0.22
End Function
```

Funzione che restituisce la lunghezza di una stringa

```
Function LunghezzaTesto(testo As String) As Integer
    LunghezzaTesto = Len(testo)
End Function
```

Esercizi

Scrivi una funzione che restituisce la somma di tre numeri

Usa tre parametri numerici e restituisci il totale.

Crea una funzione che calcola il voto medio tra tre materie

Utilizza tre input e restituisci la media.

Scrivi una funzione che restituisce TRUE se il testo contiene la parola 'Excel'

Utilizza la funzione InStr.

Crea una funzione che restituisce l'iniziale maiuscola di un nome

Utilizza Left e UCase.

Scrivi una funzione che verifica se un anno è bisestile

Utilizza la logica: divisibile per 4, ma non per 100 a meno che non sia anche divisibile per 400.

2. Esempio semplice di funzione VBA personalizzata

a. Esempio semplice

Un esempio classico di funzione personalizzata in VBA è una funzione che restituisce il doppio di un numero.

```
Function Raddoppia(numero As Double) As Double
    Raddoppia = numero * 2
End Function
```

Questa funzione prende un numero come input e restituisce il suo doppio.

b. Creazione di una funzione personalizzata

Per creare una funzione personalizzata:

1. Premi Alt + F11 per aprire l'Editor VBA
2. Inserisci un nuovo Modulo dal menu Inserisci → Modulo
3. Scrivi il codice della funzione nel modulo
4. Chiudi l'editor e torna a Excel

Assicurati di salvare il file come .xlsm per conservare la macro.

c. Utilizzo della funzione in un foglio di lavoro

Dopo aver creato la funzione, puoi usarla direttamente in una cella come una qualsiasi funzione Excel. Ad esempio: =Raddoppia(5) restituirà 10.

Verifica che il file sia salvato in formato .xlsm e che le macro siano abilitate.

d. Analisi della funzione personalizzata

Questa funzione è semplice e utile per introdurre la logica delle UDF. Può essere estesa per includere controlli sull'input, commenti per documentazione e miglioramenti nella gestione degli errori.

Ad esempio, puoi aggiungere un controllo per gestire valori negativi o nulli:

```
Function Raddoppia(numero As Double) As Variant
    If IsMissing(numero) Or numero = 0 Then
        Raddoppia = "Valore non valido"
    Else
        Raddoppia = numero * 2
    End If
End Function
```

Esempi pratici

Funzione che restituisce il triplo di un numero

```
Function Triplica(x As Double) As Double
    Triplica = x * 3
End Function
```

Funzione che somma due numeri

```
Function SommaDueNumeri(a As Double, b As Double) As Double
    SommaDueNumeri = a + b
End Function
```

Funzione che restituisce TRUE se il numero è negativo

```
Function ÈNegativo(n As Double) As Boolean
    ÈNegativo = (n < 0)
End Function
```

Funzione che calcola il quadrato di un numero

```
Function Quadrato(n As Double) As Double
    Quadrato = n ^ 2
End Function
```

Funzione che converte gradi Celsius in Fahrenheit

```
Function CelsiusAFahrenheit(c As Double) As Double
    CelsiusAFahrenheit = (c * 9 / 5) + 32
End Function
```

3. Esercizi Esecuzione delle procedure Function in VBA

a. Esecuzione delle procedure Function

Le procedure Function in VBA restituiscono un valore e possono essere eseguite sia da una cella Excel che da un'altra Sub o Function. Sono utili per calcoli, elaborazioni e logiche personalizzate che devono restituire un risultato.

b. Chiamare funzioni personalizzate da una procedura

È possibile utilizzare una Function all'interno di una Sub per sfruttare il risultato della funzione.

Esempio:

```
Sub EsempioChiamata()  
    Dim risultato As Double  
    risultato = Raddoppia(5)  
    MsgBox "Risultato: " & risultato  
End Sub
```

```
Function Raddoppia(n As Double) As Double  
    Raddoppia = n * 2  
End Function
```

c. Utilizzare funzioni personalizzate in una formula del foglio di lavoro

Dopo aver creato una funzione personalizzata, puoi usarla come qualsiasi altra funzione Excel. Esempio:
=Raddoppia(A1)

La funzione deve essere definita in un modulo standard e il file deve essere salvato come .xlsm con macro abilitate.

Esempi pratici

Funzione che calcola la potenza di un numero

```
Function Potenza(base As Double, esponente As Double) As Double  
    Potenza = base ^ esponente  
End Function
```

Uso della funzione Potenza in una Sub

```
Sub CalcolaPotenza()  
    Dim risultato As Double  
    risultato = Potenza(2, 3)  
    MsgBox "2 elevato a 3 è " & risultato  
End Sub
```

Funzione che restituisce TRUE se un numero è primo

```
Function ÈPrimo(n As Integer) As Boolean  
    Dim i As Integer  
    If n < 2 Then ÈPrimo = False: Exit Function  
    For i = 2 To Sqr(n)  
        If n Mod i = 0 Then ÈPrimo = False: Exit Function  
    Next  
    ÈPrimo = True  
End Function
```

Uso della funzione ÈPrimo in una cella

Inserisci =ÈPrimo(7) in una cella per verificare se 7 è primo.

Funzione che calcola l'area del cerchio

```
Function AreaCerchio(r As Double) As Double  
    AreaCerchio = 3.14159 * r ^ 2  
End Function
```

Esercizi

Scrivi una funzione che calcola il perimetro di un rettangolo

Utilizza base e altezza come parametri.

Crea una Sub che richiama una funzione per convertire euro in dollari

Assumi un tasso fisso come parametro interno alla funzione.

Usa una funzione per restituire TRUE se un testo contiene la parola 'Excel'

Testa l'utilizzo sia in codice VBA che direttamente in cella.

Scrivi una funzione che restituisce la somma di una gamma di celle

Utilizza il tipo Range come parametro e somma i valori.

Crea una Sub che richiama la funzione AreaCerchio e visualizza il risultato

Mostra il risultato in una MsgBox.

Crea una funzione che calcola l'area di un rettangolo

Prendi lunghezza e larghezza come parametri e restituisci l'area.

Scrivi una funzione che verifica se un numero è divisibile per 5

Utilizza l'operatore Mod per la verifica.

Crea una funzione che restituisce la prima lettera di un testo

Utilizza la funzione Left del VBA.

Scrivi una funzione che restituisce la radice quadrata di un numero

Utilizza la funzione Sqr disponibile in VBA.

Crea una funzione che restituisce il valore assoluto di un numero

Utilizza la funzione Abs.

4. Utilizzo degli Argomenti nelle Procedure Function in VBA

a. Utilizzo degli argomenti nelle funzioni

Le funzioni VBA possono accettare argomenti (parametri) che influenzano il comportamento della funzione. Gli argomenti possono essere obbligatori o opzionali, singoli valori o intervalli.

b. Creare una funzione senza argomenti

```
Function DataCorrente() As Date
    DataCorrente = Date
End Function
```

Questa funzione restituisce la data corrente senza richiedere alcun input.

c. Creare una funzione con un argomento

```
Function Quadrato(x As Double) As Double
    Quadrato = x ^ 2
End Function
```

L'utente fornisce un numero e la funzione restituisce il suo quadrato.

d. Creare un'altra funzione con un argomento

```
Function Iniziale(testo As String) As String
    Iniziale = UCase(Left(testo, 1))
End Function
```

Restituisce la lettera iniziale maiuscola della stringa.

e. Creare una funzione con due argomenti

```
Function SommaDue(a As Double, b As Double) As Double
    SommaDue = a + b
End Function
```

Accetta due numeri e restituisce la loro somma.

f. Creare una funzione con un argomento di tipo Range

```
Function SommaIntervallo(celle As Range) As Double
    Dim c As Range
    For Each c In celle
        If IsNumeric(c.Value) Then
            SommaIntervallo = SommaIntervallo + c.Value
        End If
    Next c
End Function
```

Questa funzione calcola la somma dei valori numerici in un intervallo.

g. Creare una funzione semplice ma utile

```
Function ÈVuota(cella As Range) As Boolean
    ÈVuota = IsEmpty(cella.Value)
End Function
```

Restituisce TRUE se la cella è vuota, altrimenti FALSE.

Esempi pratici

Funzione che converte un valore in euro a dollari

```
Function EuroADollari(euro As Double) As Double
    EuroADollari = euro * 1.1
End Function
```

Funzione che restituisce TRUE se il numero è pari

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

Funzione che restituisce il numero massimo tra due

```
Function Massimo(a As Double, b As Double) As Double
```

```
If a > b Then Massimo = a Else Massimo = b
```

```
End Function
```

Funzione che conta le celle non vuote in un intervallo

```
Function ContaNonVuoti(rng As Range) As Integer
```

```
Dim c As Range
```

```
For Each c In rng
```

```
If Not IsEmpty(c) Then ContaNonVuoti = ContaNonVuoti + 1
```

```
Next
```

```
End Function
```

Funzione che restituisce il valore medio di un intervallo

```
Function MediaIntervallo(rng As Range) As Double
```

```
MediaIntervallo = WorksheetFunction.Average(rng)
```

```
End Function
```

Esercizi

Scrivi una funzione che calcola l'area di un triangolo

Prendi base e altezza come parametri e restituisci l'area.

Crea una funzione che restituisce TRUE se una stringa contiene la parola 'Excel'

Utilizza la funzione InStr.

Scrivi una funzione che calcola la differenza assoluta tra due numeri

Utilizza la funzione Abs.

Crea una funzione che restituisce la lunghezza totale di testo in un intervallo

Somma le lunghezze di ogni cella nel range.

Scrivi una funzione che restituisce il valore più piccolo in un intervallo

Utilizza WorksheetFunction.Min.

5. Debug e Inserimento di Funzioni Personalizzate in VBA

a. Debug delle funzioni personalizzate

Il debug delle funzioni VBA (User Defined Functions) è fondamentale per assicurarsi che restituiscano il risultato corretto e non causino errori durante l'esecuzione nei fogli di lavoro. Puoi usare strumenti integrati come:

- F9 per impostare un punto di interruzione
- F8 per eseguire il codice passo-passo
- La finestra Immediate per stampare variabili (Debug.Print)

Esempio:

```
Function SommaDue(a As Double, b As Double) As Double
    Debug.Print "a = " & a & ", b = " & b
    SommaDue = a + b
End Function
```

Usa Debug.Print per monitorare il valore delle variabili durante l'esecuzione.

b. Inserimento di funzioni personalizzate

Una funzione personalizzata deve essere inserita in un modulo standard:

1. Premi Alt + F11 per aprire l'Editor VBA
2. Inserisci → Modulo
3. Scrivi la funzione nel modulo
4. Salva il file come .xlsm

Una volta creata, la funzione sarà disponibile nei fogli Excel come qualsiasi funzione nativa.

Esempio di utilizzo in cella: =SommaDue(3; 5)

Esempi pratici

Funzione con Debug.Print

```
Function Quadrato(n As Double) As Double
    Debug.Print "Valore ricevuto: " & n
    Quadrato = n ^ 2
End Function
```

Uso di MsgBox per il debug

```
Function Doppio(x As Double) As Double
    MsgBox "Input ricevuto: " & x
    Doppio = x * 2
End Function
```

Inserimento di una funzione in un modulo

Alt + F11 → Inserisci → Modulo → incolla la funzione e chiudi l'editor.

Funzione che segnala errore con If

```
Function Divisione(a As Double, b As Double) As Variant
    If b = 0 Then
        Divisione = "Errore: Divisione per zero"
    Else
        Divisione = a / b
    End If
End Function
```

Uso della Finestra Immediate

Scrivi: Debug.Print Variabile durante il debug e premi Ctrl + G per vedere il risultato.

Esercizi

Crea una funzione che calcola la somma e stampa i valori passati

Usa Debug.Print per visualizzare a e b.

Inserisci una funzione che restituisce il cubo di un numero

Verifica l'esecuzione passo-passo con F8.

Scrivi una funzione con gestione errore per la radice quadrata

Controlla se il valore è negativo e restituisci messaggio d'errore.

Utilizza MsgBox per mostrare il contenuto di una stringa passata alla funzione

Verifica se è vuota o contiene testo.

Inserisci una funzione in un nuovo modulo e usala in un foglio

Prova a utilizzarla con diversi input e verifica i risultati.