

FUNZIONI PERSONALIZZATE

Le funzioni sono una serie di istruzioni con un nome unico, le quali possono essere attivate diverse volte usando il loro nome e passando gli eventuali valori di input (parametri)



- Ogni funzione ha un nome
- Ogni funzione prende in ingresso un insieme di parametri e restituisce un valore
- Gli argomenti devono essere indicati nell'ordine prestabilito
- Il numero di argomenti è variabile

Le funzioni sono quindi "**formule predefinite**" cioè una serie di istruzioni predisposte che ci aiutano nella composizione delle nostre formule, evitandoci di scrivere tutta una lunga serie di istruzioni, inserendo semplicemente il nome della Funzione preposta ad un determinato calcolo (INT(), RADQ(), RESTO(), SOMMA() ...).

FUNZIONI PERSONALIZZATE

Con il VBA è possibile creare delle **Funzioni personalizzate**.

Le funzioni create saranno liberamente disponibili sui fogli di lavoro, proprio come le funzioni predefinite di Excel, ovviamente nei fogli che sono disponibili nella stessa cartella dove la funzione risiede. Una volta chiuso il file, la nostra funzione non sarà più disponibile in Excel (è possibile ovviare anche a questo), fino a quando non riapriremo la cartella che contiene la funzione.

COME SI SCRIVE UNA FUNZIONE

Per imparare a creare una funzione personalizzata con Excel dovremo implementare il nostro programma di calcolo con un'applicazione chiamata **VBA** che è un acronimo che sta per **Visual Basic Applications**.

Per dichiarare una funzione si scrive:

```
Function Nome(param1 As tipo, param2 As tipo) as tipo-restituito  
    'corpo della funzione  
End Function
```

Nel "corpo della funzione" si inseriscono le istruzioni che saranno eseguite quando si chiama la funzione.

Esempio 1: la funzione restituisce la stringa "Ciao a tutti"
(Formula senza parametri)

```
Function Ciao() as string
    Ciao="Ciao a Tutti"
End Function
```

nella cella basta scrivere =Ciao() e il risultato è "Ciao a Tutti"

Esempio 2:
(Formula con parametri)
codice:

```
Function Ciao_Amico(Nome as string) as string
    Ciao_Amico="Ciao a " & Nome
End Function
```

nella cella mi basta scrivere =Ciao_Amico("carlo") e il risultato è "Ciao a Carlo"

Esempio 3:

```
Function Ciao_Amico(Nome as string,Anni as integer) as string
    Ciao_Amico="Ciao a " & Nome & " (hai " & anni & " anni)"
End Function
```

nella cella mi basta scrivere
=Ciao_Amico("carlo";10)
Oppure
=Ciao_Amico(A1;A2) o =Ciao_Amico(Foglio1!A1;10)

Esempio 4: questa funzione somma 1 al valore passato

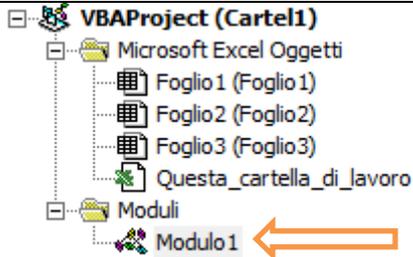
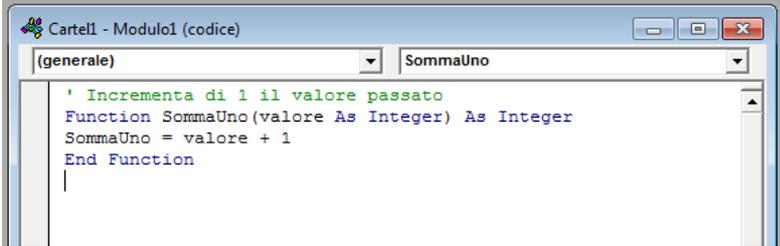
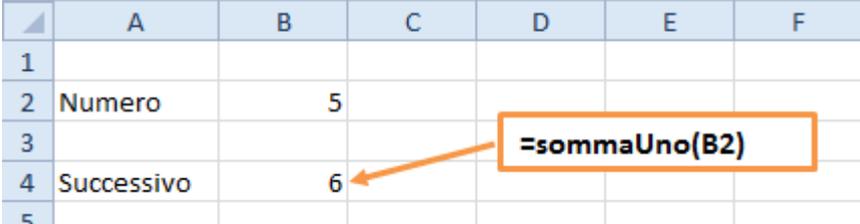
```
' Incrementa di 1 il valore passato
Function SommaUno(valore As Integer) As Integer
    SommaUno=valore+1
End Function
```

Prima della definizione è utile inserire una intestazione, come commento, per indicare lo scopo della funzione.

CREARE UNA FUNZIONE PERSONALIZZATA

Nell'esempio seguente verranno dettagliati i passaggi per la costruzione della semplice funzione AreaCerchio(Raggio) la quale consente di calcolare l'area di un cerchio a partire dal suo unico argomento: il raggio.

Per...	... usare ...
Aprire l'editor di Visual Basic	<i>Strumenti > Macro > Visual Basic Editor</i> (in alternativa: Alt + F11)
Inserire un modulo	<i>Inserisci > Modulo</i>

	
Inserire la nuova funzione	<p>Doppio click su “modulo1” e poi digitare la funzione</p> 
Tornare ad excel	File > Chiudi e torna a Microsoft Excel (in alternativa: Alt + Q)
Utilizzare la funzione	

Operatori di Confronto

<	Minore
>	Maggiore
=	Uguale
<=	Minore o uguale
>=	Maggiore o uguale
<>	diverso

Operatori matematici

+	addizione
-	sottrazione
*	Moltiplicazione
/	Divisione
\	Divisione con restituzione della sola parte intera
^	Elevamento a potenza
Mod	Resto della divisione

Operatori Logici

Se dovessimo combinare due o più espressioni, attraverso uno o più operatori relazionali, abbiamo anche gli **operatori logici**. Essi li riassumiamo in questa tabella:

Not	Negazione - Il risultato del test $A > B$ se dovesse essere uguale a true; anteporre Not a quell'espressione significherebbe eseguire una negazione. Sarebbe come effettuare il test $a <= b$		
Or	Disgiunzione logica - Abbiamo due espressioni da testare. Seguendo questa tabella di verità sapremo quando il risultato dell'applicazione della disgiunzione sarà Vero		
	$A > B = \text{VERO}$	$B > C = \text{VERO}$	$A > B \text{ OR } B > C = \text{VERO}$
	$A > B = \text{FALSO}$	$B > C = \text{VERO}$	$A > B \text{ OR } B > C = \text{VERO}$
	$A > B = \text{VERO}$	$B > C = \text{FALSO}$	$A > B \text{ OR } B > C = \text{VERO}$
	$A > B = \text{FALSO}$	$B > C = \text{FALSO}$	$A > B \text{ OR } B > C = \text{FALSO}$
And	Congiunzione Logica - Abbiamo due espressioni da testare. Seguendo questa tabella di verità sapremo quando il risultato dell'applicazione della congiunzione logica sarà Vero		
	$A > B = \text{VERO}$	$B > C = \text{VERO}$	$A > B \text{ AND } B > C = \text{VERO}$
	$A > B = \text{FALSO}$	$B > C = \text{VERO}$	$A > B \text{ AND } B > C = \text{FALSO}$
	$A > B = \text{VERO}$	$B > C = \text{FALSO}$	$A > B \text{ AND } B > C = \text{FALSO}$
	$A > B = \text{FALSO}$	$B > C = \text{FALSO}$	$A > B \text{ AND } B > C = \text{FALSO}$