



## Variabili e tipi di dati

Prima di addentrarci nell'argomento della lezione faremo una panoramica su come vengono memorizzati vari tipi di informazioni come numeri, date e testi, in modo che possano essere distinti tra di loro e come possano essere manipolati ed elaborati. Per chiarire meglio l'argomento e cercare di capire di cosa stiamo parlando è opportuno occuparsi ora delle diverse tipologie di dati che può trattare VBA e anche dello spazio che questi dati occupano in memoria. Faremo ora alcune considerazioni sulla distinzione dei tipi di dato (tipo di un dato è il termine che fa riferimento alla particolare natura dei dati che il VBA può memorizzare e manipolare quali testo e numeri) e cercheremo di vedere come può un programma lavorare con dati di tipo diverso come stringhe (la stringa è una sequenza di caratteri di testo) e numeri.

Possiamo comprendere meglio quanto esposto e la sua importanza con un esempio. Prendiamo una sequenza di caratteri "17081974" a prima vista sembra trattarsi di un numero, ma potrebbe anche rappresentare una data, il 17 Agosto del 1974 oppure potrebbe essere un numero di telefono (170 81 974). Come facciamo allora a determinare che cosa rappresenta quella stringa?

Possiamo determinare cosa rappresenta la stringa in base all'uso che ne dobbiamo fare del dato. Se si tratta di un importo sicuramente usato per eseguire dei calcoli, di conseguenza si tratta di un numero, se invece rappresenta una data verrà utilizzata in modo particolare, si può sommare ma va trattata in modo diverso oppure se è un numero di telefono sappiamo che non può essere trattato per il calcolo anche se è costituito da una sequenza numerica

Da queste considerazioni nasce l'esigenza o la necessità di dover sapere sempre la natura e lo scopo di un dato all'interno di un programma. I vari tipi di dati possono essere di tipo: **Booleano, Byte, Date, String, Integer, Single, Long, Double, Currency, Variant**, I più utilizzati sono:

- Byte : E' di tipo numerico intero senza segno compreso nell'intervallo da 0 a 255
- Integer : E'usato per rappresentare numeri interi (con segno) compresi fra -32768 a 32767
- String : Può contenere delle sequenze di caratteri (stringhe) a lunghezza variabile oppure a lunghezza fissa
- Long : Rappresenta numeri interi compresi fra -2,147,483,648 e 2,147,483,647,
- Single e Double : Utilizzato per memorizzare numeri reali a singola o a doppia precisione
- Boolean : E' di tipo logico che può assumere il valore TRUE o FALSE (vero o falso)
- Date : E' utilizzato per memorizzare data e ora
- Variant : E' un tipo universale che può contenere dei dati di qualsiasi formato

Questi vari tipi di dati vengono memorizzati da VBA per poterli utilizzare come sotto forma di *variabile* che consentono di memorizzare temporaneamente dei valori durante l'esecuzione di un'applicazione. Alle variabili deve essere associato un nome, utilizzato per fare riferimento al valore della variabile, e un tipo di dati che determina la modalità di memorizzazione dei bit che rappresentano i valori nella memoria del computer.

### ■ Le Variabili

Possiamo dire che VBA memorizza i vari tipi di dati in un'area di memoria del computer usata per contenere ogni tipo di dati, immaginiamo che una variabile sia come una casella in cui si può inserire un dato di qualsiasi tipo e salvarlo per impiegarlo successivamente. Il nome della variabile è l'etichetta che identifica la casella e il contenuto della casella è il valore della variabile, la particolarità di una variabile è di poter cambiare il suo valore durante l'esecuzione della macro, mentre il nome rimane inalterato. Possiamo quindi sintetizzare che una variabile VBA è il nome assegnato ad una specifica locazione di memoria del computer, e possiamo usare il nome della variabile per riferirci a qualsiasi dato contenuto in quella determinata locazione di memoria.



Il nome di una variabile deve essere scelto seguendo poche regole

- Deve cominciare con una lettera dell'alfabeto
- Dopo la prima lettera può contenere qualsiasi combinazione di numeri, lettere
- Il nome di una variabile non può contenere spazi, punti o caratteri speciali quali =, +, -, / e simili.
- Il nome della variabile non deve corrispondere a parole chiave di VBA
- Il nome di una variabile deve essere unico, cioè non può essere duplicato all'interno di un modulo

Creare una variabile, è abbastanza semplice, da quanto abbiamo esposto finora basta solo dargli un nome ed assegnargli un valore, vediamo un esempio.

*alex = 10*

Questo enunciato memorizza il *valore 10* nella locazione di memoria denominata *alex*, se si tratta del primo enunciato VBA crea la variabile, riserva una locazione di memoria per contenere il dato della variabile e poi memorizza il valore 10 in questa nuova locazione di memoria specificato dal nome della variabile. Se la variabile *alex* esiste già VBA memorizza il nuovo valore nella locazione di memoria a cui fa riferimento la variabile *alex*, in pratica sovrascrive il valore. Questa procedura è definita una "*dichiarazione implicita*", oppure "*dichiarazione al volo*", risulta molto comoda ma può presentare degli inconvenienti, infatti usando il metodo implicito la variabile creata da VBA è di tipo *Variant* (tutti i tipi di dati), inoltre se in seguito digitiamo il nome sbagliato (es. *Alex*), a seconda del punto in cui il nome sbagliato compare nel codice il VBA può generare un *errore di runtime*, oppure possiamo anche usare la sintassi corretta, ma così andremmo a distruggere il valore memorizzato precedentemente.

Allora come possiamo ovviare a questi inconvenienti? Dichiarando le variabili. La dichiarazione delle variabili è definita "*Dichiarazione esplicita*" e presenta i seguenti vantaggi

- Rende più veloce l'esecuzione del codice
- Aiuta ad evitare errori di digitazione
- Il codice risulta più facile da leggere
- Normalizza l'uso delle maiuscole nel nome delle variabili, per esempio se dichiariamo la variabile come *Alex* e in seguito digitiamo *alex* VBA trasforma automaticamente *alex* in *Alex*.

Per dichiarare esplicitamente una variabile si usa la parola chiave **Dim** in questo modo

**Dim nome\_variabile** che nel nostro esempio diventa *Dim alex*

E' indubbio che dichiarando le variabili ne ricaviamo notevoli benefici, ma l'errore umano nella digitazione del codice è sempre in agguato, per tutelarsi ulteriormente possiamo inserire un'altra parola chiave **Option Explicit**, se aggiungiamo questa parola chiave nell'area delle dichiarazioni di un modulo, cioè all'inizio del modulo prima di qualsiasi altra dichiarazione o listato, il VBA ci richiede di dichiarare tutte le variabili tramite l'enunciato *Dim* prima di usarle, in pratica l'enunciato *Option Explicit* proibisce di dichiarare implicitamente variabili in ogni punto del modulo, possiamo dire che con l'istruzione *Option Explicit* abbiamo aggiunto un altro pezzettino al nostro listato per garantirne una perfetta esecuzione.

Una dichiarazione implicita contiene dati di tipo *Variant*, però il nostro obbiettivo è quello di abbinare le potenzialità di VBA per utilizzare o manipolare vari tipi di dati presenti nel nostro foglio di Excel, se per esempio volessimo eseguire una somma tra i dati contenuti in due variabili avremmo sicuramente una incompatibilità nei dati e quasi certamente ci verrà rimandato un errore. Per ovviare a questo ultimo inconveniente usiamo un'altra parola chiave nella dichiarazione della variabile e aggiungiamo anche il tipo di dati che andrà a contenere. La parola chiave è **As** e l'enunciato si presenta in questo modo:

**Dim nome\_variabile As tipo** che nel nostro esempio diventa così *Dim alex As Integer*



Così facendo abbiamo creato una variabile di nome *alex* e abbiamo dichiarato che è di tipo numerico. Inoltre è possibile dichiarare più variabili usando queste sintassi

*Dim Alex As Integer, Dim x As String, Dim y As Date*, oppure in un unico blocco

*Dim Alex As Integer*

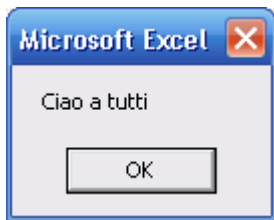
*Dim x As String*

*Dim y As Date*

Vediamo un esempio

```
Sub var1()  
Dim alex As String  
alex = "Ciao a tutti"  
MsgBox alex  
End Sub
```

E otteniamo un messaggio del genere

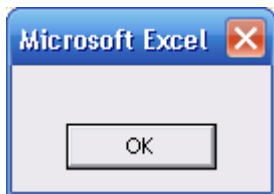


**Fig. 1**

In questo modo abbiamo dichiarato la variabile all'interno della routine o sub *var1* e può essere usata solo in quella routine, infatti se usiamo questo codice

```
Sub var1()  
Dim alex As String  
alex = "Ciao a tutti"  
stampa_box  
End Sub  
  
Private Sub stampa_box()  
MsgBox alex  
End Sub
```

E otteniamo un messaggio del genere



**Fig. 2**

Non ci viene rimandato un errore, in quanto il comando `MsgBox` viene eseguito, ma non vediamo nessuna scritta, cioè la variabile *alex* non viene riconosciuta e non appare nel nostro box.

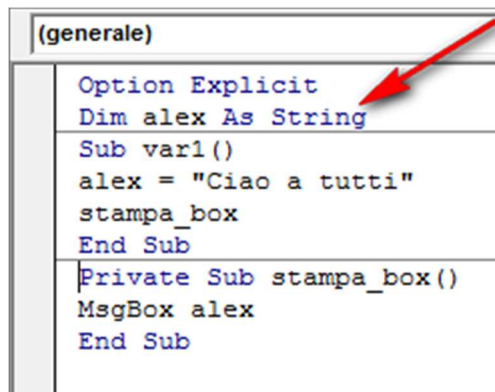


Abbiamo parlato poco sopra di dichiarazione delle variabili nell'area di dichiarazione del modulo, assieme alla parola chiave *Option Explicit*, abbiamo anche già visto i benefici di questa particolare procedura, ma se in quell'area aggiungessimo anche la dichiarazione della variabile cosa succederebbe? Semplicemente che la variabile sarebbe condivisa e utilizzabile da tutte le routine di quel modulo. Vediamo un esempio modificando il codice del listato sopra esposto

```
Option Explicit
Dim alex As String
Sub var1()
alex = "Ciao a tutti"
stampa_box
End Sub

Private Sub stampa_box()
MsgBox alex
End Sub
```

Il listato va inserito nell'editor all'inizio del modulo come mostrato in [figura 3](#)



**Fig. 3**

Se eseguiamo questa macro verrà mostrato il contenuto della variabile in una finestra

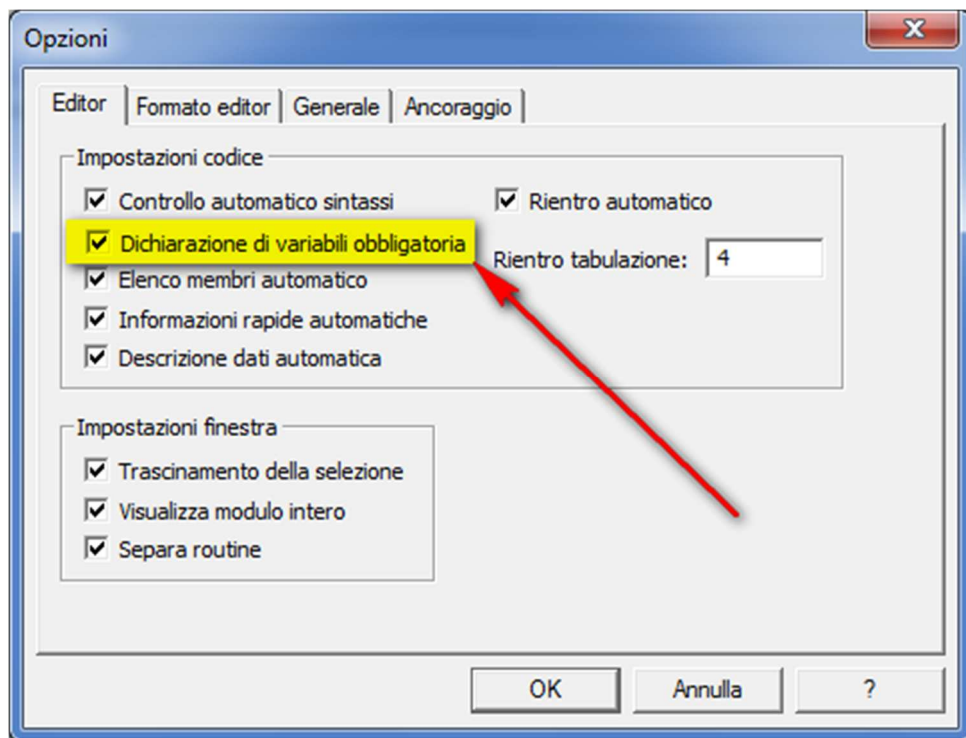


**Fig. 4**

Avrete notato che nella routine principale è stato inserito il nome di un'altra routine cioè *stampa\_box*, in questa forma la macro riconosce che quella è una chiamata ad un'altra macro e la esegue, inoltre la routine *stampa\_box* è preceduta dalla funzione **Private**, che viene usata quando vogliamo utilizzare una routine di quel modulo e solo in quello, al tempo stesso questa routine non ci compare nella finestra di assegnazione delle macro.



Abbiamo visto l'utilità nell'inserire la parola chiave *Option Explicit* nel listato, è possibile evitare di inserire in ogni modulo tale riga di codice agendo nelle opzioni dell'editor per garantire che *Option Explicit* sia sempre inserito nella parte superiore del modulo operando in questo modo: dal menu **Strumenti - Opzioni** e nella finestra che ci viene mostrata mettere il flag alla voce Dichiarazione di variabili obbligatoria. Come mostrato in [figura 5](#)



**Fig. 5**

Messa la spunta cliccare sul tasto Ok per confermare. È ora necessario utilizzare sempre la parola chiave *Dim* per dichiarare una variabile, in caso contrario verrà rimandato un errore di "*Variabile non definita*". Abbiamo anche visto l'uso della funzione *Private*, quando facciamo programmi con listati lunghi e usiamo diversi moduli, ne facilita l'interpretazione del codice e il debug in caso di errore.

Abbiamo visto che tutte le variabili dichiarate all'interno di una procedura sono disponibili solo all'interno della procedura in cui le dichiarate, mentre quelle che dichiarate a livello di modulo (come in [Fig. 3](#)) sono disponibili a tutte le procedure all'interno del modulo in cui sono state dichiarate, ma *non sono disponibili* a procedure in un modulo diverso. In VBA gli elementi che sono disponibili a tutti i moduli vengono definiti a validità pubblica e sono chiamati variabili globali, perché sono disponibili globalmente cioè in tutto il vostro programma tramite la parole chiave **Public** usando la seguente sintassi

*Public NomeVariabile [As NomeTipo]*

Dove *NomeVariabile* rappresenta un nome valido qualsiasi per identificare la variabile e *NomeTipo* un qualsiasi nome di tipo di dato valido per fare un esempio vediamo il listato di prima modificandone la dicitura con la parola chiave *Public* come mostrato in [figura 6](#)



```
(generale)  
  
Option Explicit  
Public alex As String  
  
Sub var1()  
alex = "Ciao a tutti"  
stampa_box  
End Sub  
  
Private Sub stampa_box()  
MsgBox alex  
End Sub
```

**Fig. 6**

Utilizzare Public per dichiarare una variabile a livello globale può essere utile quando si ramifica il programma in diversi moduli ma è da usare con molta attenzione specialmente nell'assegnazione del nome alla variabile per evitare di creare confusione se esistono due variabili con lo stesso nome in moduli diversi, pertanto cercate di essere espliciti nell'uso della variabile nella forma Public