



## Variabili e tipi di dati: Nozioni di Base

In programmazione, una variabile è un valore che viene affidato al computer per memorizzarlo temporaneamente nella sua memoria mentre il programma è in esecuzione. Si deve tener presente che la memoria del computer è suddivisa in piccole aree di stoccaggio utilizzate per contenere i valori delle applicazioni e quando si utilizza un valore nel codice, il computer lo memorizza in una di queste aree per poi rilasciarlo quando viene richiamato.

### ■ Dichiarazione di una variabile

Durante la scrittura del codice, è possibile utilizzare qualsiasi variabile semplicemente specificandone il nome, ed è possibile utilizzare qualsiasi nome per una variabile, ma per eliminare la possibilità di fare confusione, si deve innanzitutto far sapere al VBA che si prevede di utilizzare una variabile, al fine di prenotare l'area di stoccaggio. Per compiere questa operazione si deve dichiarare la variabile che viene utilizzata tramite la parola chiave **Dim**, inoltre una variabile deve avere un nome che deve essere posto sul lato destro della parola chiave Dim. Ci sono delle regole da seguire quando si nominano le variabili:

- Il nome di una variabile deve iniziare con una lettera o un carattere di sottolineatura
- Il nome può essere composto da lettere, sottolineature, e cifre in qualsiasi ordine
- Il nome di una variabile può contenere fino a 255 caratteri.
- Il nome di una variabile deve essere unico nella zona in cui viene utilizzato
- Alcune parole non possono essere utilizzate, in quanto sono riservate ad uso interno del VBA.

Come già detto, per dichiarare una variabile, si usa la parola chiave Dim seguita da un nome. Ecco un esempio:

```
Sub Test ()  
    Dim nome_variabile  
End Sub
```

La dichiarazione di una variabile comunica semplicemente al VBA il nome della stessa ma è comunque possibile utilizzare un mix di variabili dichiarate e non dichiarate. Se si dichiara una variabile e poi si inizia ad utilizzare un'altra variabile con un nome simile, per Visual Basic si stanno utilizzando due variabili e questo può creare confusione. La soluzione a questo problema è di dire a Visual Basic che una variabile non può essere utilizzata se non è stata dichiarata e per ottenere questo, basta inserire la parola chiave **Option Explicit** all'inizio del listato. Questa operazione può anche essere fatta automaticamente per ogni file controllando che sia inserita la spunta alla voce **Richiedi dichiarazione di variabili** nella finestra di dialogo **Opzioni** dell'editor di VB

### ■ Dichiarazione di molte variabili

In una normale applicazione, non è raro dover utilizzare molte variabili e si dovrebbe prendere l'abitudine di dichiarare sempre una variabile prima di utilizzarla. Per dichiarare una nuova variabile dopo averne dichiarato una prima, si può semplicemente andare alla riga successiva e utilizzare la parola chiave Dim per dichiarare quella nuova. Ecco un esempio:

```
Sub Test ()  
    Dim pippo  
    Dim pluto  
End Sub
```



Allo stesso modo, è possibile dichiarare quante variabili vogliamo, inoltre è possibile dichiarare più variabili sulla stessa riga e per effettuare questa operazione, si utilizza sempre la parola chiave Dim separando i nomi delle variabili con una virgola. Ecco un esempio

```
Sub Test ()  
    Dim padre, madre  
    Dim figlio, figlia, nipote, zio  
    Dim nonna  
End Sub
```

### ■ Assegnazione di valori

Abbiamo visto che quando si dichiara una variabile, il computer gli riserva uno spazio di memoria, ma lo spazio è tenuto vuoto, solo dopo aver dichiarato il valore, è possibile memorizzarlo nella memoria che è stata riservata. Per memorizzare un valore nella memoria riservata a una variabile, è possibile assegnare un valore alla variabile e per effettuare questa operazione, si digita il nome della variabile, seguito dal simbolo di assegnazione (= uguale) e seguito dal valore che si desidera memorizzare. Ecco un esempio:

```
Sub Test ()  
    Dim Valore  
    Valore = 9374  
End Sub
```

Ci sono diversi tipi di valori che si possono utilizzare nel documento, inoltre, il valore assegnato alla variabile deve essere in accordo con il tipo di memoria che il computer aveva riservato, pertanto solo dopo l'assegnazione di un valore ad una variabile, è possibile utilizzarla conoscendone il valore. In qualsiasi momento e, se necessario, è possibile modificare il valore contenuto nella variabile, è per questo che si chiama variabile (perché il suo valore può variare o cambiare) e per modificare il valore contenuto, si deve accedere nuovamente alla variabile e assegnare il nuovo valore desiderato.

### ■ Tipi di dati

Un tipo di dati indica al computer che tipo di variabile si intende utilizzare perché prima di utilizzare una variabile, si dovrebbe sapere quanto spazio occuperà in memoria. Diverse variabili utilizzano diverse quantità di spazio in memoria e le informazioni che specificano la quantità di spazio di cui ha bisogno la variabile è chiamato tipo di dati e viene misurato in **byte**. Per specificare il tipo di dati che verrà utilizzato per una variabile, dopo aver digitato la parola chiave Dim seguito dal nome della variabile si deve digitare la parola chiave **As** seguita da uno dei tipi di dati che esamineremo successivamente. La formula utilizzata è: *Dim nome\_variabile As tipo\_dati*

Abbiamo detto in precedenza che è possibile utilizzare diverse variabili se sono necessarie e quando si dichiara tali variabili, abbiamo visto che si potevano dichiarare su righe separate e per specificare il tipo di dati si utilizza la stessa formula di cui sopra. Ecco un esempio:

```
Sub Test ()  
    Dim Nome As tipo_dati  
    Dim Cognome As tipo_dati  
End Sub
```



Abbiamo anche visto che si potrebbe dichiarare varie variabili sulla stessa riga a patto che siano separate con una virgola, ora se si specifica il tipo di dati di ogni variabile si usa la stessa regola, si digita la virgola dopo ogni variabile. Ecco alcuni esempi:

```
Sub Test ()  
    Dim Nome As tipo_dati, Cognome As tipo_dati  
    Dim Indirizzo As tipo_dati, Città As tipo_dati, Nazione As tipo_dati  
    Dim Sesso As tipo_dati  
End Sub
```

Il codice sopra esposto appare come se vi fosse un solo tipo di dati, di seguito passeremo in rassegna i vari tipi di valori che sono disponibili, e per dichiarare le variabili di tipi di dati diversi, si dichiara ognuna su una riga come abbiamo visto in precedenza:

```
Sub Test ()  
    Dim Nome_Cognome As tipo_dati1  
    Dim Data_nascita As tipo_dati2  
End Sub
```

È inoltre possibile dichiarare variabili di tipi di dati diversi sulla stessa linea e per fare questo, si utilizza sempre la parola chiave Dim separando le dichiarazioni con le virgole. Ecco un esempio:

```
Sub Test ()  
    Dim Nome_Cognome As tipo_dati1, Data_nascita As tipo_dati2  
End Sub
```

### ■ Tipo di Caratteri

Per rendere la dichiarazione della variabile più veloce e anche conveniente, è possibile sostituire l'espressione As tipo\_dati con un carattere speciale che rappresenta il tipo di dati previsto. Questo carattere si chiama *Tipo di carattere* e dipende dal tipo di dati che si intende applicare a una variabile e se viene utilizzato, il tipo di carattere deve essere l'ultimo carattere del nome della variabile. Vedremo quali caratteri sono disponibili e quando possono essere applicati

### ■ Valore di conversione

Ogni volta che l'utente inserisce un valore in un'applicazione, tale valore viene considerato in primo luogo come testo, ciò significa che, se si desidera utilizzare tale valore in un'espressione o un calcolo che prevede un valore specifico diverso dal testo, è necessario convertire quel testo. Fortunatamente, Microsoft Visual Basic fornisce un meccanismo efficace per convertire un valore di testo di uno degli altri valori che vedremo dopo. Per convertire il testo ad un altro valore, vi è una parola chiave adatta allo scopo e che dipende dal tipo di valore in cui si desidera convertirlo.

### ■ Variabili numeriche - Integer

Se vogliamo utilizzare un numero nel programma, Visual Basic è in grado di riconoscere un numero naturale qualsiasi che non include una parte frazionaria e il numero è costituito da una combinazione di 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9, nessun altro carattere è consentito.



## ■ Byte

Per dichiarare una variabile che conterrebbe numeri naturali che vanno da 0 a 255, si utilizza come tipo di dati **Byte**. Ecco un esempio:

```
Sub Test ()  
    Dim AnniServizio As Byte  
End Sub
```

Non esiste un tipo di carattere per il tipo di dati Byte e dopo aver dichiarato la variabile, è possibile assegnare un piccolo numero positivo. Ecco un esempio:

```
Sub Test ()  
    Dim Valore As Byte  
    Valore = 246  
End Sub
```

È inoltre possibile utilizzare il numero in formato esadecimale fino a quando il numero è inferiore a 255 e se si dà un valore negativo o un valore superiore a 255, quando si tenta di accedervi, si riceverà un errore. Per convertire un testo in un numero piccolo, è possibile utilizzare la funzione **CByte ()** utilizzando la seguente formula: *Numero = CByte (Valore da convertire a Byte)* si ricorda che quando si utilizza CByte (), si deve immettere il valore da convertire tra parentesi.

## ■ Integer

Per dichiarare una variabile che potrebbe contenere un numero che varia -32.768 a 32.767, si utilizza come tipo di dati **Integer**. Ecco un esempio di dichiarazione di una variabile intera:

```
Sub Test ()  
    Dim conta As Integer  
End Sub
```

Invece di utilizzare l'espressione *As Integer*, è possibile utilizzare il carattere **%** come tipo di dati, pertanto, la dichiarazione di cui sopra può essere fatta come segue:

```
Sub Test ()  
    Dim conta %  
End Sub
```

Dopo aver dichiarato la variabile, è possibile assegnarle il valore desiderato, se si assegna un valore inferiore a -32768 o superiore a 32767, quando si decide di usarla, si riceverà un errore. Se si vuole convertire un testo in un numero, si può usare la funzione **CInt ()** utilizzando la seguente formula: *Numero = CInt (Valore da convertire)* dove tra le parentesi di CInt () si deve inserire il testo, o l'espressione che deve essere convertita.

## ■ Long

Un intero Long è un numero che può essere utilizzato per una variabile che coinvolge un numero maggiore di Integer, pertanto per dichiarare una variabile che potrebbe contenere un numero così elevato, si utilizza il tipo di dati **Long**. Ecco un esempio:



```
Sub Test ()  
    Dim Popolazione As Long  
End Sub
```

Il tipo di carattere per il tipo di dati Long è @, pertanto la variabile di cui sopra può essere dichiarata come segue:

```
Sub Test ()  
    Dim Popolazione @  
End Sub
```

Una variabile Long può memorizzare un valore compreso tra -2,147,483,648 e 2,147,483,647 (le virgole sono usate per facilitare la lettura, non devono essere utilizzati nel codice), pertanto, dopo aver dichiarato una variabile Long, è possibile assegnare un numero in tale intervallo. Per convertire un valore testo in un intero Long, si usa l'espressione **CLng ()** utilizzando la seguente formula: *Numero = CLng (Valore da convertire)*, ricordando di inserire nelle parentesi di CLng () il testo da convertire

#### ■ Variabili decimali a Singola precisione - Single

Nella programmazione, un numero decimale è quello che rappresenta una frazione, esempi sono 1.85 e 426,88. Se si prevede di utilizzare una variabile che sarebbe di quel tipo di numero, ma la precisione non è la vostra preoccupazione principale, si dichiara come tipo di dati **Single**. Ecco un esempio:

```
Sub Test ()  
    Dim Distanza As Single  
End Sub
```

Il tipo di carattere per il tipo di dati Single è ! pertanto la dichiarazione di cui sopra potrebbe essere scritta in questo modo:

```
Sub Test ()  
    Dim Distanza !  
End Sub
```

Se si dispone di un valore testo che deve essere convertito, si utilizza la funzione **CSng ()** con la seguente formula: *Numero = CSng (Valore da convertire)*, nelle parentesi di CSng () immettere il valore da convertire.

#### ■ Variabili decimali a Doppia precisione - Double

Se si desidera utilizzare un numero decimale che richiede una buona dose di precisione, si dichiara una variabile con tipo di dati **Double**. Ecco un esempio:

```
Sub Test ()  
    Dim Distanza As Double  
End Sub
```



Invece di *As Double*, è possibile utilizzare il tipo di carattere **#**

```
Sub Test ()  
    Dim Distanza #  
End Sub
```

Per convertire un valore testo in Double si utilizza la funzione **CDbl ()** con la seguente formula: *Numero = CDbl (Valore da convertire)* inserendo nelle parentesi di CDbl (), il valore che deve essere convertito.

### ■ String

Una stringa è un carattere o una combinazione di caratteri che costituiscono il testo di qualsiasi tipo e quasi qualsiasi lunghezza. Per dichiarare una variabile di tipo stringa, si utilizza come tipo di dati **String**. Ecco un esempio:

```
Sub Test ()  
    Dim Paese As String  
End Sub
```

Il tipo di carattere per i dati String è **\$**, pertanto, la dichiarazione di cui sopra può essere scritta come:

```
Sub Test ()  
    Dim Paese $  
End Sub
```

Come già detto, dopo aver dichiarato una variabile, è possibile assegnarle un valore, nel caso di una variabile stringa il valore deve essere incluso all'interno di doppi apici. Ecco un esempio:

```
Sub Test ()  
    Dim Paese As String  
    Paese = "Italia"  
End Sub
```

Se si dispone di un valore che non è in formato testo e si desidera convertirlo in una stringa, si deve utilizzare la funzione **CStr ()** con la seguente formula: *CStr (valore da convertire in stringa)*, inserendo nelle parentesi di CStr (), il valore che si desidera convertire in stringa.

### ■ Currency - Valuta

Il tipo di dati valuta viene utilizzato per trattare valori monetari. Ecco un esempio di dichiarazione:

```
Sub Test ()  
    Dim Salario As Currency  
End Sub
```

Invece di utilizzare la valuta come espressione, è possibile utilizzare il carattere **@** come tipo di dati per dichiarare una variabile di valuta. Ecco un esempio:

```
Sub Test ()  
    Dim Salario @  
End Sub
```





Quando si assegna un valore a una variabile **Currency** non si deve utilizzare il simbolo di valuta. Ecco un esempio di assegnazione di un numero di valuta per una variabile:

```
Sub Test ()  
Dim Salario As Currency  
    Salario = 66500  
End Sub
```

Se si desidera convertire un valore in Currency, si deve utilizzare la funzione **CCur ()** con la seguente formula: *Number = CCur (Valore da convertire)*, inserendo il valore da convertire tra le parentesi di CCur ().

### ■ Date - Data

Una data come tipo di dati può essere utilizzata per memorizzare un valore data, pertanto, per dichiarare una variabile data si utilizza come tipo di dati **Date**. Ecco un esempio:

```
Sub Test ()  
    Dim Data_nascita As Date  
End Sub
```

Dopo aver dichiarato la variabile, è possibile assegnare un valore, che deve essere compreso tra due simboli # (cancelletto). Ecco un esempio:

```
Sub Test ()  
    Dim Data_nascita As Date  
    Data_nascita = # 10/8/1988 #  
End Sub
```

Se si dispone di una stringa o un'espressione che si desidera convertire in un valore data, si deve utilizzare la funzione **CDate ()** in base alla seguente formula: *Risultato = CDate (Valore da convertire)*, inserendo nelle parentesi di CDate (), il valore che deve essere convertito.

### ■ Time

In Visual Basic, il tipo di dati Date può essere utilizzato anche per memorizzare un valore di tempo, ecco un esempio di dichiarazione di una variabile che può contenere un valore di tempo:

```
Sub Test ()  
    Dim tempo As Date  
End Sub
```

Per assegnare un valore alla variabile si segue la sintassi della funzione Date



## ■ Variant

Fino ad ora, abbiamo dichiarato variabili conoscendone il tipo di valore che dovevano contenere, VBA offre un tipo di dati universale che è possibile utilizzare per qualsiasi tipo di valore, il tipo di dati Variant. Questo tipo di dati viene utilizzato per dichiarare una variabile il cui contenuto non è esplicitamente specificato, ciò significa che un tipo di dati **Variant** può contenere qualsiasi tipo di valore che si desidera. Ecco alcuni esempi di variabili di tipo Variant, che contengono diversi tipi di valori:

```
Sub Test ()  
    Dim Nome As Variant  
    Dim Paese As Variant  
    Dim Salario As Variant  
    Dim DataN As Variant  
    Nome = "Eva Kant"  
    Paese = 2  
    Salario = 35.65  
    DataN = # 24/02/2004#  
End Sub
```

Finora nelle variabili che abbiamo dichiarato, abbiamo sempre specificato un tipo di dati, è possibile dichiarare una variabile senza il suo tipo di dati. Ecco alcuni esempi:

```
Sub Test ()  
    Dim Nome As Variant  
    Dim Paese As Variant  
    Dim Salario As Variant  
    Dim DataN As Variant  
End Sub
```

Naturalmente, è possibile dichiarare più di una variabile sulla stessa linea e per indicare la quantità di spazio necessaria per la variabile, è necessario assegnare un valore. Ecco alcuni esempi:

```
Sub Test ()  
    Dim Nome As Variant  
    Dim Paese As Variant  
    Dim Salario As Variant  
    Dim DataN As Variant  
    Nome = "Eva Kant"  
    Paese = 2  
    Salario = 35.65  
    DataN = # 24/02/2004#  
End Sub
```

## ■ Durata di una variabile

Fino ad ora, abbiamo dichiarato le variabili tra le linee di codice Sub e End Sub, questo tipo di variabile è definita come variabile locale. Una variabile locale è limitata alla zona in cui è dichiarata, in pratica non è possibile utilizzare tali variabili al di fuori della Sub in cui è stata dichiarata. Ecco un esempio:





```
Option Explicit
Sub Test ()
    Dim Nome As String
    Nome = "Eva"
End Sub
```

### ■ Variabili globali

Una variabile globale è una variabile dichiarata al di fuori della Sub e questo tipo di variabile è normalmente dichiarata nella sezione superiore del file. Ecco un esempio:

```
Option Explicit
Dim Cognome As String

Sub Test ()
    .....
End Sub
```

Dopo aver dichiarato una variabile globale, è possibile accedervi dalle altre aree del file. Ecco un esempio:  
In Modulo1

```
Option Explicit
Dim Cognome As String
```

In Modulo2

```
Sub Test ()
    Dim Nome As String
    Nome = "Eva"
    Cognome = "Kant"
End Sub
```

Anche se abbiamo dichiarato la nostra variabile globale all'interno del file in cui è stato utilizzato, è anche possibile dichiarare una variabile globale in un modulo separato per essere in grado di utilizzarlo in un altro modulo.

### ■ Variabili private

Una variabile è indicata come Private se può accedere solo al codice all'interno dello stesso modulo, dove viene utilizzato. Per dichiarare una tale variabile, invece della parola chiave Dim, si utilizza la parola chiave **Private**. Ecco un esempio:

```
Option Explicit
Private Cognome As String

Sub Test ()
    Dim Nome As String
    Nome = "Eva"
    Cognome = "Kant"
End Sub
```



Ricordate che una variabile privata è accessibile da qualsiasi codice nello stesso modulo

### ■ Variabili pubbliche

Una variabile è denominata **Public** se si può accedere al codice sia all'interno dello stesso modulo in cui è dichiarata o dal codice esterno relativo modulo. Per dichiarare una variabile pubblica, invece della parola chiave Dim, si utilizza la parola chiave **Public**. Ecco un esempio:

```
Option Explicit
Private Cognome As String
Public Nome_esteso As String

Sub Test ()
    Dim Nome As String
    Nome = "Eva"
    Cognome = "Kant"
    Nome_esteso = Nome & "" & Cognome
End Sub
```

Come promemoria possiamo affermare che una variabile pubblica è disponibile per codificare dentro e fuori dal suo modulo, ciò significa che è possibile creare un modulo, dichiarare una variabile pubblica, e accedere a quella variabile da un altro modulo, mentre una variabile privata è disponibile all'interno del suo modulo, ma non al di fuori dello stesso. Se si dichiara una variabile privata in un modulo e si prova ad accedervi da un altro modulo, si riceverà un errore:

#### Modulo 1

```
Option Explicit
Private Cognome As String
```

#### Modulo 2

```
Option Explicit
Private Cognome As String
Private Nome As String

Sub Test ()
    Nome = "Eva"
    Cognome = "Kant"
    Nome_esteso = Nome & "" & Cognome
    ActiveCell.FormulaR1C1 = Nome_esteso
End Sub
```