



Le Funzioni Empty – ZLS – Null – Nothing e Missing

In Excel VBA spesso ci riferiamo a una variabile *Empty* (vuota), a una *ZLS* (stringa di lunghezza zero) o a una stringa nulla (*vbNullString*), a un valore *Null* o a un argomento mancante (*Missing*) o utilizzando la parola chiave *Nothing* (Niente) con una variabile oggetto. E' importante differenziare e comprendere questi termini ed espressioni mentre vengono utilizzati nel codice VBA. Vediamo ora di comprendere come utilizzare la funzione *VarType* per determinare il sottotipo di una variabile, utilizzando le funzioni *IsEmpty* e *IsNull* per verificare la presenza di valori vuoti, e la funzione *IsMissing* per verificare se gli argomenti opzionali sono stati elencati nella procedura.

■ La Funzione Empty

Quando si dichiara una variabile utilizzando un'istruzione Dim, si sta riservando la parte sufficiente di memoria per allocare la variabile nel sistema, (cioè 2 byte per una variabile booleana o Integer, 4 byte per una variabile Long, e così via), e inoltre ci si deve accertare che le informazioni che saranno memorizzate nella variabile abbiano un intervallo consentito (True o False per una variabile booleana, un numero intero compreso tra -32.768 e 32.767 per una variabile Integer, un numero intero compreso tra -2.147.483.648 a 2.147.483.647 per una variabile Long, e così via).

Quando in una dichiarazione di una variabile non si specifica il tipo di dati, oppure se non viene dichiarata, assumerà per impostazione predefinita la forma di tipo **Variant** e può contenere qualsiasi tipo di dati (stringa, data, ora, booleano o valori numerici) e sarà in grado di convertire automaticamente i valori che contiene. Tuttavia, lo svantaggio di questa assegnazione è che deve essere riservata più memoria di quanto richiesto (almeno 16 byte), oltre al fatto che in caso di un errore di digitazione di un nome di variabile non saremmo in grado di riconoscerlo, vale a dire che è possibile digitare *rowNumbre* invece di *rowNumber*.

Quando si esegue una macro, tutte le variabili vengono inizializzate ad un valore predefinito e Il valore di default iniziale per una variabile numerica è zero, per una stringa di lunghezza variabile è una lunghezza zero o stringa vuota (""), una stringa di lunghezza fissa viene inizializzata con il codice ASCII 0, o Chr (0), una variabile oggetto su Nothing e una variabile Variant viene inizializzata a vuoto. Nel contesto numerico, una variabile vuota indica uno zero, mentre in un contesto di una variabile stringa vuota è una stringa di lunghezza zero ("") che è indicata anche come una stringa nulla. Tuttavia, si consiglia di specificare esplicitamente un valore iniziale per una variabile, invece di basarsi sul suo valore iniziale di default.

■ La Funzione IsEmpty

Si può utilizzare la funzione **IsEmpty** per controllare se una variabile è stata inizializzata, in questo caso la funzione restituisce un valore booleano, restituisce True per una variabile non inizializzata o se una variabile è impostata in modo esplicito a Empty, altrimenti la funzione restituisce False. La sintassi è la seguente: *IsEmpty (espressione)*, dove *espressione* è una variabile di tipo Variant che si desidera controllare.

■ La Funzione ZLS (stringa di lunghezza zero) o vbNullString

ZLS significa stringa di lunghezza zero (""), ed è indicata anche come una stringa nulla, e ha una lunghezza pari a zero (0). Per tutti gli scopi pratici si può utilizzare la costante **vbNullString** che è equivalente a una stringa di lunghezza zero (""), perché VBA interpreta in un modo simile, anche se entrambi non sono in realtà la stessa cosa, in quanto una stringa di lunghezza zero significa in realtà la creazione di una stringa senza caratteri, mentre *vbNullString* è una costante utilizzata per un puntatore nullo il che significa che nessuna stringa viene creata ed è anche più efficiente o più veloce da eseguire rispetto a ZLS. È possibile usare "" o *vbNullString* in alternativa nel codice ed entrambi si comportano allo stesso modo, si noti che non vi è alcuna parola chiave Empty in VBA, ma possiamo fare riferimento a "celle vuote "o "celle vuote nel foglio di calcolo Excel".



■ La Funzione VarType

Si utilizza la funzione **VarType** per determinare il tipo di variabile con la seguente sintassi: *VarType (nome_variabile)* e restituisce un intero che indica il sottotipo della variabile. L'espressione nome_variabile può essere qualsiasi variabile, tranne un tipo di dati definito dall'utente utilizzando l'istruzione Type. Esempi di valori di ritorno sono:

Il valore 0 (costante VarType - vbEmpty, non inizializzato di default), *il valore 1* (costante VarType - vbNull, non contiene dati validi), *il valore 2* (costante VarType - vbInteger, Integer), *il valore 3* (costante VarType - vbLong, Intero long), e così via. Le costanti VarType possono essere utilizzate ovunque nel codice al posto dei valori effettivi.

Esempio: Rappresentare una variabile vuota:

```
Sub Prova1 ()  
    'la variabile Var1 non è stata dichiarata, quindi è di tipo Variant  
    'e restituisce 0, che indica come sottotipo una variabile vuota  
    MsgBox VarType (var1)  
    'restituisce True, che indica una variabile vuota  
    MsgBox IsEmpty (var1)  
    'restituisce False, è una variabile vuota, non una variabile Null  
    MsgBox IsNull (var1)  
    'una variabile vuota o uguale a zero in VBA viene rappresentata sia come uno zero  
    'che come una stringa di lunghezza zero, e restituisce entrambi i messaggi  
    If var1 = 0 Then  
        MsgBox "Variabile vuota rappresentata come Zero"  
    End If  
    If var1 = "" Then  
        MsgBox "Variabile vuota rappresentata come Zero-Length (Null) String"  
    End If  
End Sub
```

Esempio: Test per variabili vuote

```
Sub Prova2 ()  
    Dim var1 As Variant  
    'variabile non inizializzata, restituisce 0, che indica una variabile vuota  
    MsgBox VarType (var1)  
    'restituisce True, indicando una variabile vuota  
    MsgBox IsEmpty (var1)  
    'Inizializzare la variabile come stringa  
    var1 = "Ciao"  
    'restituisce 8, che indica una variabile Stringa  
    MsgBox VarType (var1)  
    'restituisce False, che indica che la variabile non è vuota  
    MsgBox IsEmpty (var1)  
    'si imposta la variabile vuota  
    var1 = Empty  
    'restituisce 0, che indica variabile vuota  
    MsgBox VarType (var1)  
    'restituisce True, che indica variabile vuota  
    MsgBox IsEmpty (var1)  
    'Restituisce True per una cella del foglio di lavoro vuota, altrimenti False  
    MsgBox IsEmpty (ActiveCell)  
End Sub
```



Esempio: Inizializzare una variabile Variant

```
Sub Prova3 ()  
Dim var1 As Variant  
'variabile inizializzata con una stringa di lunghezza zero ("")  
var1 = ""  
'restituisce False, che indica che la variabile non è vuota  
MsgBox IsEmpty (var1)  
'restituisce 8, che indica una variabile Stringa  
MsgBox VarType (var1)  
If var1 = "" Then  
MsgBox "Il valore della variabile è una stringa di lunghezza zero"  
Else  
MsgBox "Il valore della variabile NON è una stringa di lunghezza zero"  
End If  
If var1 = 0 Then  
MsgBox "Il valore della variabile è zero"  
Else  
MsgBox "Il valore della variabile non è zero"  
End If  
End Sub
```

Esempio: Controllare una stringa di lunghezza zero:

```
Sub Prova4 ()  
Dim var1 As Variant  
'variabile non inizializzata, restituisce 0, che indica una variabile vuota  
'è rappresentata sia come zero (0) che con una lunghezza zero (Null)  
MsgBox VarType (var1)  
If var1 = "" Then  
MsgBox "True"  
End If  
If var1 = vbNullString Then  
MsgBox "True"  
End If  
If Len (var1) = 0 Then  
MsgBox "True"  
End If  
End Sub
```

La funzione Null

In VBA, la parola chiave **Null** viene utilizzata per indicare che una variabile non contiene dati validi e il valore che indica una variabile che non contiene dati validi il risultato è Null se:

- Si assegna esplicitamente Null a una variabile
- Se si eseguono operazioni tra espressioni che contengono la parola chiave Null

La parola chiave Null viene utilizzata per variabili di tipo Variant, e solo una variabile Variant può essere Null, mentre variabili di qualsiasi altro tipo rimanderanno un errore, inoltre una variabile Null non è da intendere come una stringa di lunghezza zero (""), e non è vuota, ma indica una variabile non ancora inizializzata. Se si tenta di ottenere il valore di una variabile Null o un'espressione che è Null, si otterrà un *errore 94 di Utilizzo non valido di Null*.



■ La Funzione IsNull

La funzione **IsNull** restituisce un valore booleano, dove True rappresenta un'espressione che Null (non contiene dati validi), mentre False indica un'espressione che contiene dati validi. La sintassi è la seguente: *IsNull (espressione)* e l'argomento *espressione* è una variante che contiene un valore numerico o stringa.

Esempio: Variabile Integer

```
Sub Prova5 ()  
Dim intVar As Integer 'nessun valore iniziale è assegnato alla variabile Integer  
MsgBox IsNull (intVar) 'Restituisce False (intVar non è Null o Empty)  
MsgBox VarType (intVar) 'restituisce 2, indicando il tipo Integer  
If intVar = 0 Then  
MsgBox "Il valore della variabile è zero"  
Else  
MsgBox "Il valore della variabile non è zero"  
End If  
End Sub
```

Esempio: Valutare se la variabile è Empty o Null

```
Sub Prova6 ()  
Dim var1 As Variant  
'restituisce False, var1 non è Null, ma è vuota  
MsgBox IsNull (var1)  
'la variabile non è inizializzata e restituisce 0, che indica una variabile vuota  
MsgBox VarType (var1)  
'restituisce il messaggio perché var1 è una variabile vuota  
If var1 = 0 And var1 = vbNullString Then  
MsgBox "Variabile vuota rappresentata sia come zero (0) che come lunghezza zero (Null) String"  
End If  
'la variabile viene inizializzata su una stringa di lunghezza zero ("" ) o vbNullString  
var1 = vbNullString  
'restituisce False, var1 non è una variabile Null  
MsgBox IsNull (var1)  
'restituisce 8, che indica una variabile stringa  
MsgBox VarType (var1)  
'si assegna Null alla variabile  
var1 = Null  
'restituisce True, una variabile Null, non contenente dati validi  
MsgBox IsNull (var1)  
'restituisce 1, indicando una variabile Null  
MsgBox VarType (var1)  
'assegnare dei dati validi alla variabile  
var1 = 12  
'restituisce False, per una variabile che contiene dati validi  
MsgBox IsNull (var1)  
'restituisce 2, indicando una variabile integer  
MsgBox VarType (var1)  
'restituisce False, per un'espressione contenente dati validi  
MsgBox IsNull ("Ciao")  
End Sub
```



Esempio: Controllare una variabile Null

```
Sub Prova7 ()  
    'si assegna Null alla variabile  
    var1 = Null  
    'restituisce 1, indicando una variabile Null  
    MsgBox VarType (var1)  
    'restituisce il messaggio, indicando una variabile Null  
    If VarType (var1) = vbNull Then  
        MsgBox "Variabile Null"  
    End If  
    'un'espressione contenente Null restituisce ancora Null  
    var2 = Null + 2  
    'restituisce 1, indicando una variabile Null  
    MsgBox VarType (var2)  
End Sub
```

Esempio: Controllare una cella del foglio di lavoro

```
Sub Prova8 ()  
    Dim var1 As Variant  
    'restituisce True  
    MsgBox vbNullString = ""  
    'se ActiveCell è vuota restituisce True  
    MsgBox ActiveCell.Value = ""  
    MsgBox ActiveCell.Value = vbNullString  
    MsgBox ActiveCell.Value = 0  
    MsgBox IsEmpty (ActiveCell.Value)  
    'assegnare il valore della cella attiva alla variabile  
    var1 = ActiveCell.Value  
    'restituisce True  
    MsgBox IsEmpty (var1)  
    MsgBox var1 = vbNullString  
    MsgBox var1 = ""  
    MsgBox var1 = 0  
    'restituisce False  
    MsgBox VarType (var1) = vbNull  
    'restituisce 0, che indica una variabile vuota  
    MsgBox VarType (var1)  
    'se si immette "" nella cella attiva restituisce True  
    MsgBox ActiveCell.Value = ""  
    MsgBox ActiveCell.Value = vbNullString  
    'restituisce False  
    MsgBox ActiveCell.Value = 0  
    MsgBox IsEmpty (ActiveCell.Value)  
End Sub
```



■ La Funzione Nothing

L'assegnazione della parola chiave **Nothing** a una variabile oggetto dissocia la variabile stessa da un oggetto reale e questa assegnazione avviene utilizzando l'istruzione Set. Abbiamo visto in precedenza che ogni assegnazione eseguita a delle variabili vengono utilizzate delle risorse di sistema per allocare in memoria la variabile. Le risorse di sistema e di memoria vengono rilasciati solo dopo aver assegnato Nothing tramite l'istruzione Set a tutte le variabili oggetto che di dissociare queste variabili dall'oggetto reale, o quando tutte le variabili oggetto vengono distrutte. *Si consiglia di impostare esplicitamente tutte le variabili oggetto a Nothing al termine della procedura o anche prima durante l'esecuzione, quando il codice ha finito di usarle, e questo rilascerà memoria allocata per queste variabili.*

Per controllare se un oggetto è stato assegnato o impostato, si utilizza la parola chiave **IsNothing**, vale a dire usando un'espressione come la seguente: *If object_variable Is Nothing*

Esempio: Utilizzare la parola chiave Nothing con una variabile oggetto

```
Sub Prova9()  
Dim OVar As Object  
'restituisce True, perché non è stato assegnato un oggetto reale alla variabile oggetto  
MsgBox OVar Is Nothing  
Set OVar = ActiveSheet  
'restituisce False, perché è stato assegnato un oggetto reale (foglio) alla variabile  
MsgBox OVar Is Nothing  
Set OVar = Nothing  
'restituisce "La variabile non è associata a un oggetto reale", perché abbiamo dissociato  
'la variabile oggetto da un oggetto reale  
If OVar Is Nothing Then  
MsgBox "La variabile non è associata a un oggetto reale"  
Else  
MsgBox "Un oggetto reale è assegnato a una variabile Object"  
End If  
End Sub
```

■ La funzione Missing

Quando un valore esterno deve essere utilizzato da una procedura per eseguire un'azione, si passa alla procedura da variabile che sono chiamati argomenti. Un argomento è il valore fornito dal codice chiamante a una procedura quando viene chiamato e quando il set di parentesi, dopo il nome della procedura nella dichiarazione Sub o Function, è vuota, si tratta di un caso in cui la procedura non riceve argomenti. Tuttavia, quando gli argomenti sono passati a una procedura da altre procedure, allora questi sono elencati o dichiarati tra le parentesi.

Gli argomenti possono essere specificati come facoltativi, utilizzando la parola chiave *Optional* prima dell'argomento alla sua sinistra e quando si specifica un argomento come opzionale, tutti gli altri argomenti successivi posti alla destra dell'argomento sono specificati come Optional. Si noti che specificando la parola chiave Optional rende un argomento opzionale altrimenti sarà richiesto l'argomento.

L'argomento opzionale dovrebbe essere (anche se non è necessario) dichiarato come tipo di dati Variant per consentire l'uso della funzione **IsMissing** che funziona solo quando viene utilizzato con le variabili dichiarate come Variant. La funzione IsMissing viene utilizzata per determinare se l'argomento opzionale è stato passato alla procedura o meno in modo che ci si può regolare di conseguenza nel codice senza restituire un errore.



Se l'argomento opzionale non è dichiarato come Variant, la funzione IsMissing non funziona, e all'argomento opzionale verrà assegnato il valore predefinito per il tipo di dati che è 0 per le variabili di tipo numerico (cioè Integer, Double, ecc) e Nothing (un riferimento nullo) per le variabili String o variabili di tipo Object.

La funzione IsMissing viene utilizzata con questa sintassi: *IsMissing (argname)* e restituisce un valore booleano, True se non viene passato nessun valore per l'argomento opzionale, e False se un valore è stato passato. Se la funzione IsMissing restituisce True per un argomento, utilizzando l'argomento mancante nel codice causerà un errore, e quindi utilizzando questa funzione aiuterà a regolare il codice di conseguenza.

Esempio: Utilizzo della funzione IsMissing per verificare se un argomento è mancante

```
Function NomeC(Pnome As String, Optional Snome As Variant) As String
'La dichiarazione della procedura contiene due argomenti, il secondo argomento è
'specificato come Optional. 'Dichiarare l'argomento opzionale come tipo di dati Variant
'consentirà l'utilizzo della funzione IsMissing.
If IsMissing(Snome) Then
NomeC = Pnome
Else
NomeC = Pnome & " " & Snome
End If
End Function

Sub Pas_Nome()
Dim nome1 As String
nome1 = InputBox("Inserire il nome")
'Specificando solo il primo argomento e omettere il secondo argomento che è facoltativo
MsgBox NomeC(nome1)
End Sub
```