



Il Metodo Find in VBA

Per cercare un articolo specifico o un valore in un intervallo, si può utilizzare il metodo *Find* che restituisce il Range, vale a dire, la cella, dove si trova l'elemento o valore. Se non viene trovata nessuna corrispondenza viene restituito *Nothing*.

Sintassi: *RangeObject.Find(What, After, LookIn, LookAt, SearchOrder, SearchDirection, MatchCase, MatchByte, SearchFormat)*

E' necessario specificare solo l'argomento *What*, tutti gli altri sono facoltativi e rappresentano:

RangeObject: Rappresenta un intervallo in cui viene cercato l'elemento o lo specifico valore ed è possibile cercare all'interno di celle specifiche, vale a dire, Range, Colonne o le celle di un foglio di lavoro.

What: E' la voce o il valore che viene ricercato e può essere di qualsiasi tipo di dati.

After: Rappresenta una singola cella che è necessario specificare, dopo di che la ricerca inizia. Perché la ricerca inizia dopo questa cella, la cella specificata viene cercata alla fine e quando la ricerca inizia dopo la cella specificata e raggiunge la fine dell'intervallo di ricerca, senza trovare il valore di ricerca, la ricerca ricomincia dall'inizio dell'intervallo di ricerca fino alla cella specificata. Se l'argomento non viene specificato, la cella iniziale è definita nell'angolo superiore sinistro del campo di ricerca, dopo di che inizia la ricerca.

Se si specifica *After*: = Range ("A13") in cui il campo di ricerca è Range ("A1: A20"), il metodo *Find* inizierà la ricerca dalla cella A14, fino alla cella A20 e successivamente cercherà dalla cella A1 fino alla cella A13.

Lookin: Questo argomento specifica il tipo di informazioni - può essere *xlValues* o *xlFormulas* o *xlComments* che indicano il tipo di valore da cercare, se una formula, un commento o un valore. Il valore predefinito è *xlFormulas*.

	A
1	12
2	24
3	87
4	78
5	98
6	82
7	176
8	57
9	176
10	90
11	76
12	323
13	432
14	45
15	234
16	somma
17	77
18	99
19	9
20	25

Fig. 1



Prendiamo il caso in cui la cella A7 di *figura 1* contenga la formula =SOMMA(A4;A5) e il totale della somma è 176, si può usare il metodo Find per cercare il valore nelle formule in questo modo:

```
ActiveSheet.Range("A1:A20").Find(What:="176", After:=ActiveSheet.Range("A1"), LookIn:=xlFormulas)
```

Verrà restituita la cella \$A\$9, perché il valore 176 NON compare nella formula della cella A7, ma compare come valore nella cella A9. Se invece usiamo il metodo Find per ricercare il valore 176 deve essere espresso in questo modo:

```
ActiveSheet.Range("A1:A20").Find(What:="176", After:=ActiveSheet.Range("A1"), LookIn:=xlValues)
```

Restituirà \$A\$7, in quanto il valore 176 appare come risultato della formula nella cella A7 e solo dopo nella cella A9. Analogamente, nel caso in cui la cella A7 contiene la formula =SOMMA(A4;A5), e la cella A16 contiene la stringa *somma*, si può applicare il metodo Find in questo modo:

```
ActiveSheet.Range("A1:A20").Find(What:="somma", After:=ActiveSheet.Range("A1"), LookIn:=xlValues)
```

Restituirà \$A\$16, mentre invece se il metodo viene espresso nel seguente modo:

```
ActiveSheet.Range("A1:A20").Find(What:="sum", After:=ActiveSheet.Range("A1"), LookIn:=xlFormulas)
```

Restituirà \$A\$7

Esempio: Le opzioni del metodo Find

```
Sub cerca_1()  
Dim valoreC As Range  
'Restituisce $A$9  
Set valoreC = ActiveSheet.Range("A1:A20").Find(What:="176",  
After:=ActiveSheet.Range("A1"), LookIn:=xlFormulas)  
If Not valoreC Is Nothing Then  
MsgBox valoreC.Address  
'End If  
'Restituisce $A$7  
Set valoreC = ActiveSheet.Range("A1:A20").Find(What:="176",  
After:=ActiveSheet.Range("A1"), LookIn:=xlValues)  
If Not valoreC Is Nothing Then  
MsgBox valoreC.Address  
'End If  
'Restituisce $A$16  
Set valoreC = ActiveSheet.Range("A1:A20").Find(What:="somma",  
After:=ActiveSheet.Range("A1"), LookIn:=xlValues)  
If Not valoreC Is Nothing Then  
MsgBox valoreC.Address  
End If  
'Restituisce $A$7  
Set valoreC = ActiveSheet.Range("A1:A20").Find(What:="sum",  
After:=ActiveSheet.Range("A1"), LookIn:=xlFormulas)  
If Not valoreC Is Nothing Then  
MsgBox valoreC.Address  
End If  
End Sub
```



E' possibile utilizzare anche gli altri parametri della sintassi del metodo Find come:

LookAt: È possibile specificare *xlWhole* o *xlPart*, se volete, rispettivamente, una corrispondenza esatta o una corrispondenza parziale. Una ricerca utilizzando *xlPart* per "Gianni" restituirà la cella che ha "Gianni Morandi", perché c'è una corrispondenza parziale. Utilizzare *xlWhole* per abbinare l'intero valore o la stringa che corrisponde esattamente al valore di una cella. Il valore predefinito è *xlPart*.

SearchOrder: È possibile specificare *xlByRows* o *xlByColumns* per questo argomento, che indicano se cercare per righe o per colonne. Il valore predefinito è *xlByRows*.

Supponiamo che le celle A7 e B3 contengano la stringa "somma" si può utilizzare il metodo Find per una ricerca nelle righe in questo modo:

```
ActiveSheet.Range("A1:B20").Find (What:= "somma", LookIn:=xlValues, SearchOrder:=xlByRows)
```

Mentre invece per una ricerca nelle colonne si deve applicare il metodo Find in questo modo:

```
ActiveSheet.Range ("A1:B20").Find (What:= "somma", LookIn: = xlValues, SearchOrder: = xlByColumns  
)
```

xlSearchDirection: È possibile specificare *xlNext* per ricerche verso il basso (cioè il valore corrispondente successivo) o *xlPrevious* per ricerche verso l'alto o all'indietro (cioè il valore corrispondente precedente) nel campo di ricerca. Il valore predefinito è *xlNext*. Se si specifica *After: = Range ("A13")* in cui il campo di ricerca è *Range ("A1: A20")* e impostare il *SearchDirection: = xlNext*, allora la funzione di ricerca inizierà a cercare dalla cella A14 fino alla cella A20 e poi ricercare dalla cella A1 fino alla cella A13.

MatchCase: Si deve specificare il valore *True* per una ricerca case-sensitive. Il valore predefinito è *False*.

MatchByte: Questo argomento può essere utilizzata solo se si seleziona il supporto delle lingue a doppio byte.

SearchFormat: Indica se si desidera cercare una specifica formattazione con il valore *True* o *False*. Il valore predefinito è *False*. Si deve specificare il formato utilizzando la proprietà *FindFormat* dell'oggetto *Application*, e impostare l'argomento *SearchFormat true*. Vedi l'esempio sotto riportato che illustra questa tesi.

Esempio: Utilizzare il metodo Find per cercare la prima occorrenza del valore stringa "somma", che è in grassetto. Si noti che la ricerca inizierà dopo la cella A1 (cioè dalla A2) in assenza dell'argomento *After*

```
Sub cerca_formato()  
Dim cerca_1 As Range, ultima As Range, cerca_val As Range  
'impostare l'intervallo di ricerca  
Set cerca_1 = ActiveSheet.Range("A1:A20")  
'specificare l'ultima cella del range  
Set ultima = cerca_1.Cells(cerca_1.Cells.Count)  
'Specificare il formato utilizzando FindFormat  
Set cerca_val = cerca_1.Find(What:="somma", After:=ultima, LookIn:=xlValues,  
SearchFormat:=True)  
MsgBox cerca_val.Address  
End Sub
```



Ogni volta che il metodo Find viene utilizzato, le impostazioni per LookIn, LookAt, SearchOrder, e MatchByte vengono salvate, a meno che i valori per questi argomenti non siano specificati di nuovo, i valori precedentemente salvati vengono utilizzati di nuovo la prossima volta che viene richiamato il metodo. Quindi è importante impostare questi argomenti **in modo esplicito** ogni volta che questo metodo viene utilizzato. In questo modo, il valore di argomento utilizzato in precedenza diventa il default se non specificato nel successivo uso.

Ad esempio, se si specifica LookIn con l'argomento xlFormulas, poi xlFormulas diventa il valore predefinito per l'argomento LookIn. E nel successivo utilizzo, se si omette di menzionare l'argomento LookIn, si imposterà xlFormulas.

Nel caso in cui si desidera cercare un elemento o un valore in un intervallo, una pratica comune è utilizzare un ciclo come Loop o For Next. Se l'intervallo di ricerca è piuttosto grande, l'uso del ciclo potrebbe richiedere molto tempo, mentre invece un grande vantaggio nell'uso del metodo Find è la sua velocità.

Per trovare occorrenze multiple di un elemento o un intervallo si può utilizzare il metodo *FindNext* o *FindPrevious*. Questi metodi sono utilizzati per proseguire la ricerca con il metodo Find, utilizzando gli stessi parametri o condizioni, e restituire il successivo (metodo FindNext) o precedente (metodo FindPrevious) cella corrispondente.

■ Il Metodo Range.FindNext: Sintassi: *RangeObject.FindNext (After)*

RangeObject: Rappresenta un intervallo in cui viene cercato l'elemento o il valore specifico.

After: Rappresenta una singola cella che si deve specificare dopo la quale inizia la ricerca. Quando la ricerca inizia dopo la cella specificata e raggiunge la fine dell'intervallo di ricerca, senza trovare il valore di ricerca, la ricerca ricomincia dall'inizio dell'intervallo di ricerca fino alla cella specificata. È facoltativo specificare questo argomento, e se non specificato, è la cella nell'angolo superiore sinistro del campo di ricerca, dopo di che inizia la ricerca.

■ Il Metodo Range.FindPrevious: Sintassi: *RangeObject.FindPrevious (After)*

RangeObject: Rappresenta un intervallo in cui viene cercato l'elemento o valore specifico.

After: Rappresenta una singola cella che si specifica **prima** che inizia la ricerca, perché la ricerca inizia da questa cella. È facoltativo specificare questo argomento, e se non specificato, è la cella nell'angolo superiore sinistro del campo di ricerca prima che inizia la ricerca.



Esempio: Trovare più occorrenze di un valore in un intervallo; trovare la stringa "vecchia" in un campo di ricerca, sostituirla con "nuova" e cambiare il colore del carattere.

```
Sub multi_cerca()  
Dim cerca1 As Range, ultima As Range, cerca_prima As Range  
Dim primo_ind As String  
'Impostare l'intervallo di ricerca  
Set cerca1 = ActiveSheet.Range("A1:A100")  
'Specifica ultima cella nel range  
Set ultima = cerca1.Cells(cerca1.Cells.Count)  
'Trovare la stringa "vecchia" nel campo di ricerca  
Set cerca_prima = cerca1.Find(What:="vecchia", After:=ultima, LookIn:=xlValues,  
LookAt:=xlWhole, SearchOrder:=xlByRows, SearchDirection:=xlNext, MatchCase:=False,  
SearchFormat:=False)  
  
'Se "vecchia" si trova nel campo di ricerca  
If Not cerca_prima Is Nothing Then  
'Salva l'indirizzo del primo risultato di "vecchia", nella variabile primo_ind  
primo_ind = cerca_prima.Address  
Do  
'Trova la successiva occorrenza di "vecchia". Si noti, che non si parte dalla prima  
occorrenza di "vecchia"  
Set cerca_prima = cerca1.FindNext(cerca_prima)  
'Sostituire "vecchia" con "nuova"  
cerca_prima.Value = "nuova"  
'Colore del carattere è cambiato  
cerca_prima.Font.Color = vbRed  
  
Loop Until cerca_prima.Address = primo_ind  
End If  
End Sub
```

Esempio: Con riferimento all'Immagine 2, sotto riportata, il codice proposto mostra come utilizzare il metodo Find e la proprietà Offset per fare una ricerca verticale.

	A	B	C	D	E	F	G
1	Inserisci Voti				Tabella Studenti		
2	Studente	Voto			Studente	Classe	Voto
3	Luca				Luca	4	55
4	Carlo	74			Carlo	3	74
5	Ivan				Ivan	3	68
6	Marco	82			Marco	5	82
7	Gianni				Gianni	4	78
8							

Fig. 2

Per ogni studente nella colonna A, si deve trovare il nome dello stesso nella colonna E, e inserire i suoi voti nella colonna B, solo se è nella classe 3



```
Sub cerca_2()  
Dim cerca1 As Range, nomeSt As Range, prima As Range, stud As Range  
Set cerca1 = ActiveSheet.Range("E3:E7")  
Set nomeSt = ActiveSheet.Range("A3:A7")  
'si ricercano tutti i nomi degli studenti menzionati nell'intervallo specificato  
For Each stud In nomeSt  
Set prima = cerca1.Find(What:=stud, LookIn:=xlValues, LookAt:=xlWhole,  
SearchOrder:=xlByRows, SearchDirection:=xlNext, MatchCase:=False,  
SearchFormat:=False)  
'Se il nome dello studente viene trovato e se studente è in classe 3  
If Not prima Is Nothing And prima.Offset(0, 1) = "3" Then  
stud.Offset(0, 1) = prima.Offset(0, 2)  
End If  
Next  
End Sub
```

■ Utilizzo del metodo Find per cercare una data

Excel memorizza tutte le date come numeri interi e il tempo come frazioni decimali. Con questo sistema, Excel può aggiungere, sottrarre o confrontare date e ore, proprio come qualsiasi altro numero, considerando come data iniziale di questo sistema il 1/1/1900 0:00:00, che Excel considera erroneamente il 1900 come un anno bisestile, si presume che questo errore sia stato fatto consapevolmente da Microsoft per garantire la compatibilità con Lotus 1-2-3, e quindi in realtà il bug sarebbe stato in Lotus 123 (predecessore di Excel).

In Excel, la data equivale a un "numero di serie" (che è un valore numerico), che è il conteggio del numero di giorni trascorsi da una certa data di riferimento. La parte intera (valori a sinistra del separatore decimale) è il numero di giorni trascorsi dal 1 gennaio 1900, ad esempio, 1 gennaio 1900 viene memorizzato come 1, il 2 gennaio 1900 viene memorizzato come 2, il 15 marzo 2001 viene memorizzato come 36.965. La parte frazionaria (valori a destra del decimale) contiene informazioni temporali, e rappresenta il tempo come frazione di un giorno intero. Ad esempio, 12:00 (mezzanotte) è memorizzato come 0, 06:00 viene memorizzato come 0,25, 12:00 (mezzogiorno) viene memorizzato come 0.5, 06:00 viene memorizzato come 0,75, 06:00:30 viene memorizzato come 0,750347222. Per controllare il "numero di serie" di una data e ora si deve formattare la cella come "Generale".

L'utilizzo del metodo Find per trovare o cercare una data può essere difficile, il formato della data deve corrispondere al formato data predefinito, come impostato nel vostro desktop, che, se non specificatamente modificato, dovrebbe essere nel suo formato standard di "data breve" o "data lunga", vale a dire, "22/01/2010" o "22 gennaio 2010". Non importa in quale data il formato viene visualizzato nel foglio di lavoro, deve solo essere una data valida per Excel corrispondente ad un numero di serie valido. Il codice seguente mostra come utilizzare il metodo Find per cercare una data



Esempio: Ricerca di una data all'interno di un intervallo

	A	B	C	D
1	2-mar-10			36254
2	10-feb-75			
3	16-set-11			
4	22-ago-10			
5	4-apr-99			
6	2-feb-85			
7				

Fig. 3

In questo esempio l'utente inserisce la data che vuole trovare nella cella D2 che viene rappresentata col valore di 36254 in quanto la cella è formattata come "Generale". La ricerca avviene nella colonna A.

```
Sub cerca_data()  
Dim primo As Range, cerca1 As Range, ultimo As Range  
Dim strDate As String  
'intervallo di ricerca  
Set cerca1 = ActiveSheet.Range("A1:A20")  
Set ultimo = cerca1.Cells(cerca1.Cells.Count)  
  
'si stabilisce che la data da cercare è in D2  
strDate = Format(ActiveSheet.Range("D1"), "Short Date")  
'Format ("4/4/99", "Short Date") restituisce "04/04/1999"  
'Format ("4/4/99", "Long Date") restituisce "Domenica 4 Aprile 1999".  
'La funzione IsDate restituisce True se l'espressione è una data valida, altrimenti restituisce False.  
If IsDate(strDate) = False Then  
MsgBox "Formato Data Incorretto"  
Exit Sub  
End If  
'CDate converte un numero o una stringa di testo in un tipo di dati Date.  
'CDate (36240) restituisce "04/04/1999"  
  
Set primo = cerca1.Find(What:=CDate(strDate), After:=ultimo, LookIn:=xlFormulas,  
LookAt:=xlWhole, SearchOrder:=xlByRows, SearchDirection:=xlNext, MatchCase:=False,  
SearchFormat:=False)  
If Not primo Is Nothing Then  
'Se la data viene trovata si stampa l'indirizzo della cella (A5 in questo esempio)  
MsgBox primo.Address  
Else  
MsgBox "Data non trovata"  
End If  
End Sub
```