



## Range, Cells e ciclo With

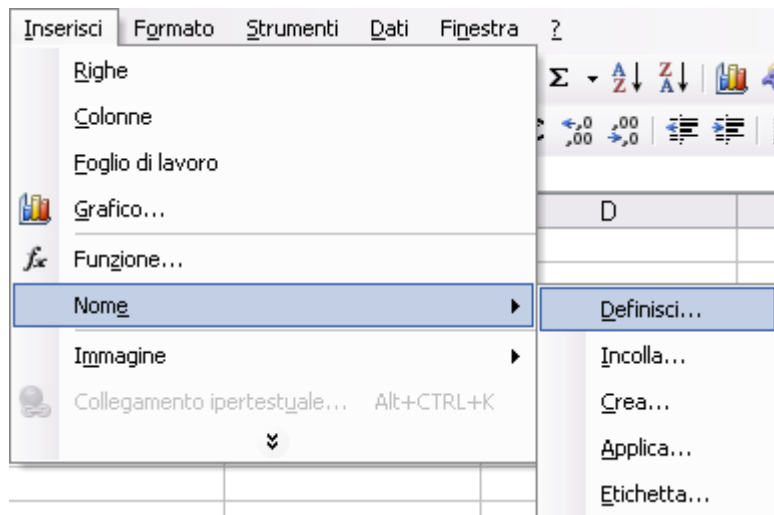
Abbiamo visto come è strutturato un foglio di calcolo o cartella, abbiamo detto che è costituito da celle, intervalli e fogli, inoltre un insieme di celle è rappresentato da righe e colonne, ricordando che la maggior parte dei compiti che svolgiamo in un foglio di calcolo è quella di introdurre informazioni, tagliare e copiare dati o applicare opzioni di formattazione e tante altre funzioni che coinvolgono celle, righe o colonne, questo insieme è definito **Range**. Un Range può essere rappresentato da una singola cella o più celle da una colonna, una riga o una selezione di celle, il sistema più facile per identificarlo è proprio il comando Range che ha questa sintassi

*Object.Range(nome)*

- **Object:** è un riferimento all'oggetto *Worksheet* che contiene il Range, se viene omissso VBA assume che si riferisca all'ActiveSheets (il foglio che abbiamo attivo)
- **Nome:** è un riferimento al Range o il nome del Range inserito come testo, infatti questo comando lavora anche con Range che hanno un nome, vediamo qualche esempio

*Worksheets("Foglio1").Range("A1").Value = 123*, Oppure possiamo rappresentarlo come un insieme di celle in questo modo: *Worksheets("Foglio1").Range("A1:C5").Value = 123*

Possiamo usarlo anche con un nome assegnato agendo in questo modo : Selezioniamo un insieme di celle e dal Menu **Inserisci - Nome - Definisci**, come da immagine sotto



**Fig. 1**



E ci comparirà un box come il seguente

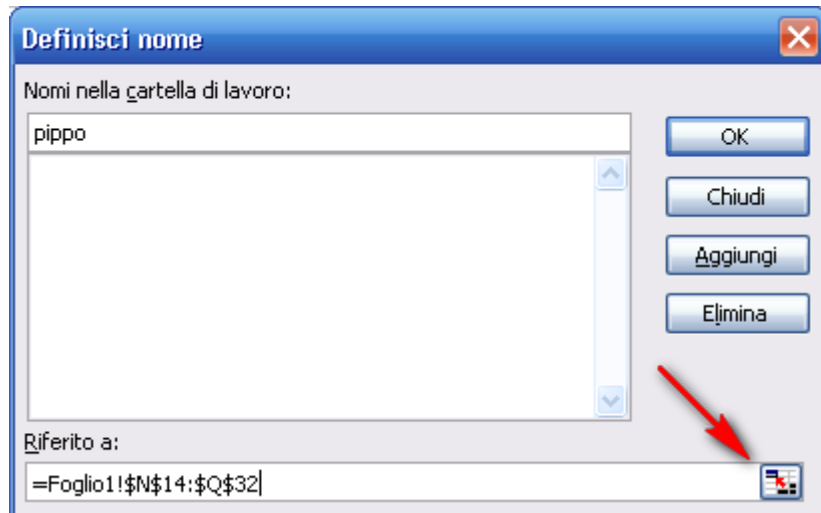


Fig. 2



**Per versioni di Excel superiori alla 2003** per assegnare un nome ad un intervallo di celle si deve seguire il percorso **Formule - Definisci nome** come da immagine sotto stante

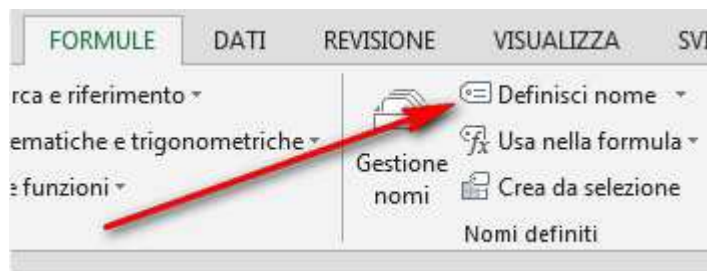


Fig. 3

E ci compare una finestra come la seguente

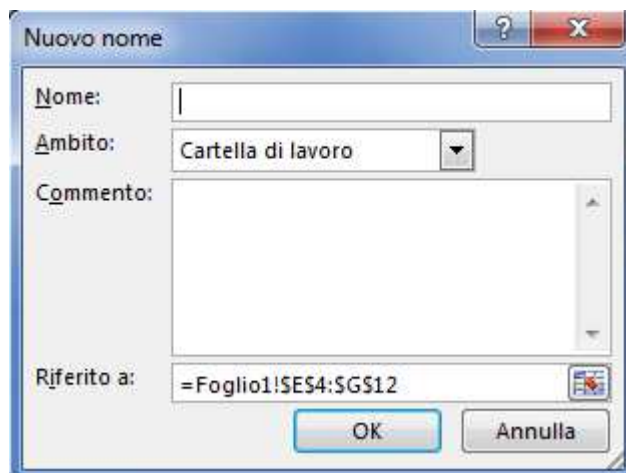
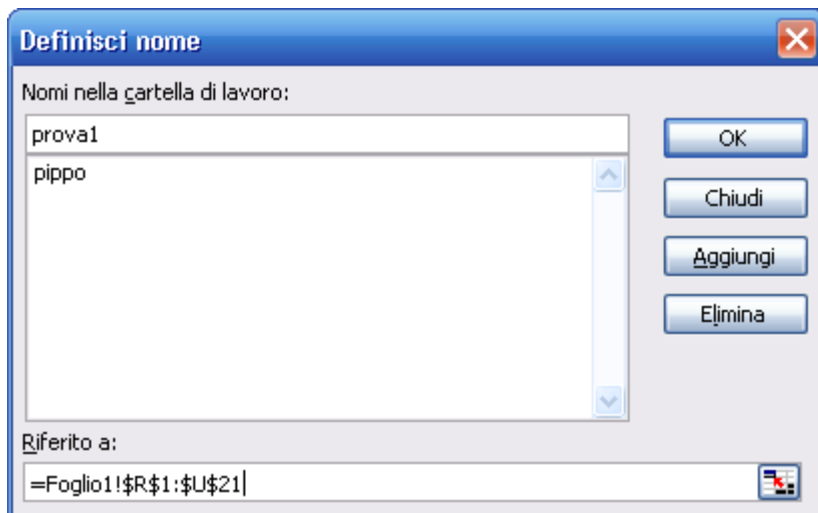


Fig. 4

**Fine Nota**



Se abbiamo selezionato prima le nostre celle troveremo il loro riferimento nel campo *Riferito a* oppure possiamo cliccare sull'icona evidenziata dalla freccia rossa e procedere alla loro selezione, fatto questo dobbiamo solo inserire il nome dell'intervallo e cliccando su *Aggiungi* lo stesso comparirà nel box centrale così:



A questo punto possiamo modificare il nostro listato in questo modo:

*Worksheets("Foglio1").Range("pippo").Value = 123*

Possiamo riepilogare che con il primo codice abbiamo riempito la cella A1 col valore 123, mentre col secondo abbiamo riempito un *insieme di celle* (dalla A1 alla C5) col valore 123, mentre assegnando un *nome ad un intervallo di celle* abbiamo riempito tutto l'intervallo col valore 123. Possiamo però dire anche che se siamo certi di operare nel foglio attivo posso omettere il riferimento *Object* e quanto abbiamo finora visto lo possiamo scrivere anche in questo modo

- *Range("A1").Value = 123* (per la sola cella A1)
- *Range("A1:C5").Value = 123* (per un intervallo di celle)
- *Range("pippo").Value = 123* (utilizzando un nome di un intervallo di celle)

Sempre sulla falsa riga di quanto appena citato possiamo ulteriormente semplificare il listato usando riferimenti assoluti in questo modo

- *[A1] = 123* (per la sola cella A1)
- *[A1:C5] = 123* (per un intervallo di celle)
- *[pippo] = 123* (utilizzando un nome di un intervallo di celle)

Come abbiamo potuto vedere il comando *Range* è estremamente flessibile e lo useremo spesso per poter interagire con Excel da VBA ora vediamo come comportarci quando dobbiamo leggere in un foglio e scrivere in un altro. Molte volte abbiamo la necessità di scrivere in un foglio i nostri dati e poterli salvare in un altro foglio per successive consultazioni, useremo sempre l'enunciato *Range*, ma gli abbineremo anche altre funzioni come *Cells* e il *ciclo With* esponiamo ora brevemente la sintassi e l'utilizzo di questi due comandi



## ■ L'enunciato Cells

Anche se possiamo usare il comando Range per riattivare una singola cella il comando **Cells** esegue lo stesso compito ma con maggior flessibilità, quando dobbiamo scrivere o leggere dati in un foglio di calcolo direttamente da VBA con l'enunciato Range siamo sempre vincolati ad una locazione ben precisa che abbiamo appena visto e denominata dal riferimento di cella, ma non sempre sappiamo dove dobbiamo leggere e scrivere, in sostanza è abbastanza difficile usare il comando Range quando dobbiamo copiare un insieme di dati scritti in varie celle estese su righe o colonne. A questo problema si può ovviare usando il comando Cells, che ha questa sintassi

*Object.Cells(riga,colonna)*

Per quanto riguarda il comando *Object* tralasciamo ulteriori spiegazioni in quanto vale quanto sopra esposto per il comando Range, noterete però che il riferimento al Range (nome) è espresso in coordinate di riga e colonna, questo ci permette di identificare una singola cella o un intervallo dalla loro posizione di riga e colonna, facciamo qualche esempio di identificazione di celle.

*A1 = Cells (1,1)*

*B5 = Cells (5,2)*

*D3 = Cells (3,4)*

Vediamo ora il ciclo With e poi sintetizziamo il tutto e uniamo i vari comandi

## Il Ciclo With

■ VBA ci fornisce questa speciale struttura **With .... End With** che ci permette di fare riferimento alle proprietà o metodi che appartengono allo stesso oggetto senza dover specificare ogni volta il riferimento completo all'oggetto, bella come esposizione tecnica ma poco chiara vero? Esponiamo la sintassi del ciclo With e poi semplifichiamo il concetto con degli esempi, la sintassi è :

*With Oggetto*

*Istruzioni*

*End With*

Chiarifichiamo ora il tutto, all'inizio del corso abbiamo parlato di Metodi e Proprietà e anche di Oggetti, abbiamo esposto l'oggetto *Workbook* (che è la cartella di lavoro cioè il nostro file), l'oggetto *Worksheet* (che è il foglio di lavoro : Foglio1, Foglio2 ecc...) e l'oggetto *Range* ( intervallo di celle, A1: B12, C1:D12, ecc...) nella definizione del ciclo *With* abbiamo detto che ci permette di omettere il riferimento completo all'oggetto (Workbook, Worksheet) quando le proprietà o i metodi che usiamo si riferiscono allo stesso oggetto, infatti basta dichiararlo una sola volta all'inizio della procedura With (vedi sintassi) così la nostra procedura risparmia il tempo che necessita per risolvere il riferimento all'oggetto per ogni proprietà o metodo all'interno dell'istruzione With. Vediamo un esempio e capirete subito come funziona, prendiamo come esempio i dati presenti in un foglio come in figura

	A	B
1		pippo
2		pluto
3		paperino
4		minni
5		topolino

**Fig. 6**



Supponiamo di trovarci nel Foglio1 e vogliamo scrivere i nostri dati nel Foglio2, per compiere questa operazione senza l'utilizzo del ciclo With dobbiamo utilizzare un listato del genere:

```
Worksheets("Foglio2").Range("B1").Value = Worksheets("Foglio1").Range("B1").Value
```

Oppure come abbiamo visto poco sopra in questo modo

```
Worksheets("Foglio2").Range("B1").Value = [B1].Value
```

```
Sheets("Foglio2").[B1].Value = [B1].Value
```

Ma così copiamo una singola cella, dobbiamo utilizzare un ciclo come abbiamo già visto in questo modo

```
Sub scrivi()  
  riga = 1  
  Do Until Sheets("Foglio1").Cells(riga, 2) = Empty  
    Sheets("Foglio2").Cells(riga, 2).Value = Sheets("Foglio1").Cells(riga, 2).Value  
    riga = riga + 1  
  Loop  
End Sub
```

Dobbiamo utilizzare una sintassi del genere in quanto dobbiamo incrementare il nostro contatore per poter scorrere tutti i dati presenti nel foglio di origine (nel nostro caso il Foglio1) e al tempo stesso incrementare la riga del foglio di destinazione, comunque già in questo listato abbiamo potuto vedere l'utilizzo del comando *Cells*, credo che sia chiaro come va utilizzato e a cosa serve, in ogni caso possiamo dire che con il comando *Cells* identifichiamo una cella ben precisa, il nostro problema sta solo nel fatto che dobbiamo dichiarare sempre il riferimento completo sia del foglio di origine che del foglio di destinazione (in presenza di un ciclo) però possiamo ovviare a tutto questo utilizzando il ciclo With in questo modo.

```
Sub scrivi_with()  
  j = 1  
  Do Until Sheets("Foglio1").Cells(j, 2) = Empty  
    With Sheets("Foglio2")  
      .Cells(j, 2) = Sheets("Foglio1").Cells(j, 2).Value  
    End With  
    j = j + 1  
  Loop  
End Sub
```

Apparentemente sembrano uguali, ma col ciclo With abbiamo evitato la dichiarazione dell'oggetto nel ciclo *Do Loop*, o meglio lo abbiamo fatto una sola volta con notevole risparmio in termini di ricerca dell'oggetto da parte di VBA e semplificando il listato in base a quanto finora citato relativo agli oggetti possiamo anche scriverlo così

```
Sub scrivi_with()  
  j = 1  
  Do Until Cells(j, 2) = Empty  
    With Sheets("Foglio2")  
      .Cells(j, 2) = Cells(j, 2).Value  
    End With  
    j = j + 1  
  Loop  
End Sub
```