



Le Proprietà Range, Cells, Item, Row e Columns

È possibile fare riferimento o accedere a un Range del foglio di lavoro utilizzando le proprietà e i metodi dell'oggetto Range che si riferisce ad una cella o a un intervallo di celle, può essere una riga, una colonna o una selezione di celle comprendenti uno o più blocchi contigui di celle. Uno degli aspetti più importanti nella codifica vba fa riferimento a intervalli all'interno di un foglio di lavoro.

■ La Proprietà Range

Abbiamo detto che un oggetto **Range** si riferisce a una cella o a un intervallo di celle e può essere una riga, una colonna o una selezione di celle comprendenti uno o più blocchi contigui di celle. Un oggetto Range fa sempre riferimento a un foglio di lavoro specifico, e Excel attualmente non supporta gli oggetti Range che si sviluppano su più fogli di lavoro. Alcuni esempi di codice:

- Oggetto Range riferito a una singola cella

```
Dim rng As Range  
Set rng = Range ("A1")
```

- Oggetto Range riferito a un blocco di celle contigue

```
Dim rng As Range  
Set rng = Range ("A1: C3")
```

- Oggetto Range riferito ad una riga

```
Dim rng As Range  
Set rng = Rows (1)
```

- Oggetto Range riferito a più colonne

```
Dim rng As Range  
Set rng = Columns ("A: C")
```

Codice per l'oggetto Range riferito a 2 o più blocchi di celle contigue, utilizzando il metodo Union e Selection

■ Metodo Union

Codice per l'oggetto Range riferito a 2 o più blocchi di celle contigue, utilizzando il metodo Union e Selection

```
Dim rng1 As Range, rng2 As Range, rngUnion As Range  
'impostare un blocco contiguo di celle come primo intervallo (o Range)  
Set rng1 = Range ("A1: B2")  
'impostare un altro blocco contiguo di celle come secondo intervallo  
Set rng2 = Range ("D3: E4")  
'assegnare una variabile (oggetto range) per rappresentare l'unione dei due intervalli  
Set rngUnion = Union (rng1, rng2)  
'colore interno impostato per l'intervallo che è l'unione di 2 oggetti Range  
rngUnion.Interior.Color = vbYellow
```



■ Proprietà Selection

'Selezionare 2 blocchi contigui di celle, utilizzando il metodo Select
Range("A1:B2,D3:E4").Select
'impostare il colore di sfondo delle celle selezionate a giallo
Selection.Interior.Color = vbYellow

■ La proprietà Worksheet

Sintassi: *WorksheetObject.Range (Cell1, Cell2)*

E' possibile utilizzare solo l'argomento *Cell1* e in questo caso dovrà essere un riferimento a un Range che può includere un operatore di intervallo (2 punti) o l'operatore di unione (virgola), o il riferimento a un intervallo che può essere un nome definito. Esempi di utilizzo di questo tipo di riferimento sono:

Worksheets ("Foglio1"). Range ("A1") che si riferisce alla cella A1, oppure

Worksheets ("Foglio1"). Range ("A1: B3"), che si riferisce alle celle A1, A2, A3, B1, B2 e B3.

Quando entrambi gli argomenti *cell1* e *cell2* vengono utilizzati (*cell1* e *cell2* sono oggetti Range), si riferiscono alle celle comprese tra l'angolo superiore sinistro e l'angolo inferiore destro del Range, cioè le celle iniziali e finali del Range, e gli argomenti possono essere una singola cella, un'intera riga o colonna o una singola cella denominata. Un esempio di utilizzo di questo tipo di riferimento è

Worksheets ("Foglio1"). Range (Cells (1, 1), Cells (3, 2)), che si riferisce alle celle A1, A2, A3, B1, B2 e B3. Utilizzando il codice *Range ("A1")* verrà restituita la cella A1 del foglio attivo, come se si utilizza la sintassi *Application.Range ("A1")* o *ActiveSheet.Range ("A1")*

■ La proprietà Range

Sintassi: *RangeObject.Range (Cell1, Cell2)*

Per accedere a un Range relativo a un intervallo, ad esempio:

Worksheets("Foglio1").Range("C5:E8").Range("A1") si farà riferimento al Range ("C5") mentre il codice *Worksheets("Foglio1").Range("C5:E8").Range("B2")* farà riferimento al Range ("D6")

Rispetto ad utilizzare la proprietà Range, è possibile utilizzare anche un codice breve per fare riferimento a un intervallo utilizzando le parentesi quadre per racchiudere un riferimento di tipo "A1" o un nome. Durante l'utilizzo delle parentesi quadre, non si racchiude l'intervallo tra virgolette per renderlo una stringa. Utilizzando le parentesi quadre è come applicare il metodo *Evaluate* dell'oggetto Application, in cui la proprietà Range o il metodo Evaluate utilizzano un argomento stringa che permette di manipolare la stringa con codice VBA

Esempi: utilizzando

[A1]. Value = 5 è equivalente all'utilizzo di

Range ("A1"). Value = 5, mentre usando

[A1: A3, B2: B4, C3: D5]. Interior.Color = vbRed equivale a

Range ("A1: A3, B2: B4, C3: D5"). Interior.Color = vbRed e con intervalli denominati come:

[pippo].Interior.Color = vbBlue equivale a

Range ("pippo"). Interior.Color = vbBlue.



Utilizzando le parentesi quadre si consentono solo riferimenti a intervalli fissi, mentre invece usando la proprietà **Range** si permette di manipolare l'argomento stringa con codice VBA in modo che è possibile utilizzare delle variabili per fare riferimento a un Range dinamico, come illustrato di seguito:

```
Sub RangeDinamico ()  
Dim i As Integer  
'si inserisce Ciao nelle celle da 1 a 5 della colonna B  
For i = 1 To 5  
Range ("B" & i) = "Ciao"  
next i  
End Sub
```

■ La proprietà **Cells**

Restituisce un oggetto **Range** riferito a tutte le celle di un foglio di lavoro o a un intervallo, in quanto può essere utilizzata con riferimento ad un oggetto **Application**, a un oggetto foglio di lavoro o un oggetto **Range**. La proprietà **Application.Cells** si riferisce a tutte le celle del foglio di lavoro attivo ed è possibile utilizzare **Application.Cells** nel codice o omettere il qualificatore di oggetto e utilizzare il codice per riferirsi a tutte le celle del foglio di lavoro attivo.

■ La proprietà **Worksheet.Cells**

Sintassi: **WorksheetObject.Cells** si riferisce a tutte le celle di un foglio di lavoro specificato, utilizzando il codice: **Worksheets ("Foglio1"). Cells** si fa riferimento a tutte le celle del foglio denominato "Foglio1".

Utilizzando la struttura **Range.Cells** ci si riferisce alle celle in un intervallo specificato -

Sintassi: **RangeObject.Cells**. Questa proprietà può essere utilizzata come **Range ("A1: B5"). Cells**, ma utilizzando le celle come nome intervallo in questo caso è irrilevante perché con o senza questa dicitura il codice farà riferimento al Range A1: B5. Per fare riferimento a una cella specifica, si deve utilizzare la **Proprietà Item** dell'oggetto **Range**, specificando la riga relativa e le posizioni delle colonne dopo la parola chiave **Cells**, vale a dire,

Worksheets ("Foglio1"). Cells.Item (2, 3) si riferisce al Range **C2** e

Worksheets ("Foglio1"). Range ("C2"). Cells (2, 3) farà riferimento al Range **E3**.

Poiché la proprietà **Item** è la proprietà predefinita dell'oggetto **Range** è possibile omettere questa istruzione utilizzando questo codice

Worksheets ("Foglio1"). Cells (2, 3), che si riferisce anche al Range C2.

Si può preferire in alcuni casi utilizzare

Worksheets ("Foglio1"). Cells (2, 3) rispetto a

Worksheets ("Foglio1"). Range ("C2")

perché le variabili di riga e colonna possono essere facilmente utilizzabili

■ **La Proprietà Item**: si deve utilizzare la proprietà **Range.Item** per restituire un intervallo come offset nell'intervallo specificato Sintassi: **RangeObject.Item (RowIndex, ColumnIndex)**. È necessario specificare l'argomento **RowIndex** mentre **ColumnIndex** è opzionale. **RowIndex** è il numero di indice della cella, partendo da 1 e crescente da sinistra a destra e poi verso il basso.

Worksheets ("Foglio1"). Cells.Item (1) o

Worksheets ("Foglio1"). Cells (1) si riferisce al Range A1, mentre

Worksheets ("Foglio1"). Cells (2) si riferisce al Range B1.



Durante l'utilizzo di un solo parametro di riferimento della proprietà Item (RowIndex), se l'indice supera il numero di colonne nell'intervallo specificato, il riferimento verrà disposto alle righe successive all'interno delle colonne del Range. Tralasciando l'oggetto qualificatore imposterà il foglio attivo. Cells (16385) che si riferisce al Range A2 del foglio attivo in Excel 2007 che ha 16384 colonne e cells (16386) si riferisce all'intervallo B2, e così via. Si noti inoltre che RowIndex e ColumnIndex sono offset e relativi nell'intervallo specificato (cioè rispetto all'angolo superiore sinistro del campo specificato). Entrambe le espressioni:

Range ("B3").Item(1) e
Range ("B3: D6").Item (1) si riferiscono al Range B3.

Il seguente codice si riferisce al Range D4:

Range ("B3: D6") Item (6) o
Range ("B3: D6"). Cells (6) o
Range ("B3: D6") (6). ColumnIndex si riferisce al numero di colonna della cella e può essere un numero che inizia con 1 o può essere una stringa che inizia con la lettera "A".

Worksheets ("Foglio1"). Cells (2, 3) e
Worksheets ("Foglio1"). Cells (2, "C") si riferiscono entrambi al Range C2 in cui RowIndex è 2 e ColumnIndex è 3 (colonna C),

Range ("C2"). Cells (2, 3) si riferisce al Range E3 nel foglio attivo, e
Range ("C2"). Cells (4, 5) si riferisce al Range G5 nel foglio attivo. Utilizzando:
Range ("C2"). Item (2, 3) e Range ("C2"). Item (4, 5) si ha lo stesso effetto e si riferiscono, rispettivamente, al Range E3:G5. Utilizzando:

Range ("C2: D3"). Cells (2, 3) e
Range ("C2: D3"). Cells (4, 5) sarà come riferirsi rispettivamente al Range E3 e al Range G5. Omettere la Voce Item esprimendo il codice in questo modo
Range ("C2: D3") (2, 3) e
Range ("C2: D3") (4, 5), ci si riferisce al Range E3 e al Range G5.

■ **La Proprietà Columns** nell'Oggetto Foglio di lavoro ha la seguente

Sintassi: *WorksheetObject.Columns* per riferirsi a tutte le colonne in un foglio di lavoro che vengono restituite come un oggetto Range. Esempio:

Worksheets ("Foglio1"). Columns restituirà tutte le colonne del foglio di lavoro, mentre
Worksheets ("Foglio1"). Columns (1), restituisce la prima colonna (colonna A) del foglio di lavoro, oppure *Worksheets ("Foglio1"). Columns ("A")* restituisce la prima colonna (colonna A) e
Worksheets ("Foglio1"). Columns ("A: C") restituisce le colonne A, B e C; e così via. Tralasciando l'oggetto qualificatore si imposterà il foglio come attivo, utilizzando la colonna (1) che restituisce la prima colonna del foglio attivo.

■ **La Proprietà Columns** dell'oggetto Range ha la seguente Sintassi: *RangeObject.Columns* e viene usato per fare riferimento alle colonne in un intervallo specificato. Esempio: per inserire un colore di sfondo nelle celle di tutte le colonne del Range specificato, cioè da B2 a D4.

Worksheets ("Foglio1"). Range ("B2: D4"). Columns.Interior.Color = vbYellow.
Oppure per Inserire il colore di fondo nelle celle della prima colonna del Range B2:B4:

Worksheets ("Foglio1"). Range ("B2: D4"). Columns (1). Interior.Color = vbGreen.



Se l'oggetto specificato contiene più zone, le colonne della prima area verranno restituite solo da questa proprietà. Prendiamo l'esempio di 2 aree nell'intervallo specificato, la prima area sarà "B2: D4" e la seconda "F3: G6", il seguente codice inserisce il colore di fondo nelle celle dalla prima colonna della prima area dalle celle B2: B4:

```
Worksheets ("Foglio1"). Range ("B2: D4, F3: G6"). Columns (1). Interior.Color = vbRed
```

■ **La Proprietà Worksheet.Rows** ha la seguente Sintassi: [WorksheetObject.Rows](#) e si riferisce a tutte le righe in un foglio di lavoro che vengono restituite come un oggetto Range si può usare

Worksheets ("Foglio1"). Rows che restituirà tutte le righe del foglio di lavoro, mentre
Worksheets ("Foglio1 "). Rows (1) restituisce la prima riga (riga uno) del foglio di lavoro e
Worksheets ("Foglio1"). Rows (3) restituisce la terza riga del foglio di lavoro e
Worksheets ("Foglio1"). Rows ("1:03") restituisce le prime 3 righe, e così via.

■ **La Proprietà Range.Rows** ha la seguente Sintassi: [RangeObject.Rows](#) fa riferimento a delle righe in un intervallo specificato, per esempio, per inserire il colore di fondo in tutte le righe specificate da B2 a D4 si usa:

Worksheets ("Foglio1"). Range ("B2: D4"). Rows.Interior.Color = vbYellow mentre invece per inserire il colore di fondo nella prima riga nel Range da B2 a D2:
Worksheets ("Foglio1"). Range ("B2: D4"). Rows (1). Interior.Color = vbGreen.

Se l'oggetto specificato contiene più zone, le righe della prima area verranno restituite solo dalla proprietà Areas. Prendiamo l'esempio di 2 aree nell'intervallo specificato, la prima area è "B2: D4" e la seconda area è "F3: G6" - il seguente codice inserisce il colore di fondo nelle celle dalla prima riga della prima area alle celle da B2 a D2:

```
Worksheets ("Foglio1"). Range ("B2: D4, F3: G6"). Rows (1). Interior.Color = vbRed.
```

■ **La Proprietà Range.Offset** ha la seguente Sintassi: [RangeObject.Offset \(RowOffset, ColumnOffset\)](#) . Entrambi gli argomenti sono facoltativi, l'argomento [RowOffset](#) specifica il numero di righe dell'intervallo specificato in cui si deve spostare, tenendo presente che valori negativi indicano uno spostamento verso l'alto e valori positivi indicano lo spostamento verso il basso, il valore di default è 0. L'argomento [ColumnOffset](#) specifica il numero di colonne dell'intervallo specificato in cui ci si deve spostare, tenendo presente che valori negativi indicano lo spostamento a sinistra e valori positivi indicano lo spostamento a destra, il valore di default è 0. Esempio:

Range ("C5"). Offset (1, 2) si sposta di 1 riga e 2 colonne e si riferisce al Range E6, mentre invece
Range ("C5: D7 "). Offset (1, -2) si sposta di 1 riga verso il basso e di 2 colonne a sinistra e si riferisce al Range (A6: B8).

■ Accedere a un intervallo

con Riferimento a una singola cella, Inserire il valore 10 nella cella A1 del foglio di lavoro denominato "Foglio1"

```
Worksheets ("Foglio1"). Range ("A1"). Value = 10  
Worksheets ("Foglio1"). Range ("A1") = 10
```




Inserire il valore di 10 nella cella C2 del foglio di lavoro attivo: *ActiveSheet.Cells (2, 3). Value = 10*

Riferimento a un intervallo di celle

Inserire il valore 10 nelle celle A1, A2, A3, B1, B2 e B3 del foglio attivo

ActiveSheet.Range ("A1: B3"). Value = 10

ActiveSheet.Range ("A1", "B3"). Value = 10

ActiveSheet.Range (Cells (1, 1), Cells (3, 2)) = 10

Inserire il valore 10 nelle celle A1 e B3 del foglio denominato "Foglio1":

Worksheets ("Foglio1"). Range ("A1, B3"). Value = 10

Impostate il colore di sfondo (rosso) per le celle B2, B3, C2, C3, D2, D3 e H7 del foglio di lavoro denominato "Foglio3"

ActiveWorkbook.Worksheets ("Foglio3"). Range ("B2: D3, H7"). Interior.Color = vbRed

Inserire il valore 10 nell'intervallo denominato "pippo" del foglio di lavoro attivo, vale a dire che è possibile assegnare un nome al Range ("B2: B3") come "pippo" per inserire 10 nelle celle B2 e B3

Range ("pippo"). Value = 10

ActiveSheet.Range ("pippo"). Value = 10

Selezionare tutte le celle del foglio di lavoro attivo:

ActiveSheet.Cells.Select

Cells.Select

Impostare il font "Times New Roman" e la dimensione del carattere a 11, per tutte le celle del foglio di lavoro attivo nella cartella di lavoro attiva

ActiveWorkbook.ActiveSheet.Cells.Font.Name = "Times New Roman"

ActiveSheet.Cells.Font.Size = 11

Cells.Font.Size = 11

Righe e Colonne

Selezionare tutte le righe del foglio di lavoro attivo

ActiveSheet.Rows.Select

Inserire il valore 10 in ogni cella della riga 2 del foglio di lavoro denominato "Foglio1"

Worksheets ("Foglio1"). Righe (2). Valore = 10

Selezionare tutte le colonne del foglio di lavoro attivo

ActiveSheet.Columns.Select

Columns.Select

Inserire il valore 10 in ogni cella della colonna numero 3 del foglio di lavoro attivo

ActiveSheet.Columns (3). Value = 10

Columns ("C"). Value = 10

Inserire il valore 10 in ogni cella delle colonne 1, 2 e 3 del foglio denominato "Foglio1"

Worksheets ("Foglio1"). Columns ("A: C"). Value = 10



■ Riferimento relativo

Inserisce il valore 10 in C5

Range ("C5: E8"). Range ("A1") = 10

Inserisce il valore 10 in D6 - riferimento inizia dall'angolo superiore sinistro del campo definito:

Range ("C5: E8"). Range ("B2") = 10

Inserisce il valore 10 in E6, con offset di 1 riga e 2 colonne

Range ("C5"). Offset (1, 2) = 10

Inserisce il valore 10 nel Range ("F7: H10") con Offset di 2 righe e 3 colonne

Range ("C5: E8"). Offset (2, 3) = 10

```
Sub esempio_1()  
Dim ws As Worksheet, rng As Range  
Dim r As Integer, c As Integer, n As Integer, i As Integer, j As Integer  
Set ws = Worksheets("Foglio1")  
ws.Activate  
For r = 1 To 5  
    n = 1  
    For c = 1 To 5  
        Cells(r, c).Value = n  
        n = n + 1  
    Next c  
Next r  
Set rng = Range(Cells(1, 1), Cells(5, 5))  
For i = 1 To 5  
    If i Mod 2 = 0 Then  
        rng.Columns(i).Interior.Color = vbYellow  
    Else  
        rng.Columns(i).Interior.Color = vbGreen  
    End If  
Next i  
  
For j = 1 To 5  
    If j Mod 2 = 0 Then  
        rng.Rows(j).Font.Bold = True  
        rng.Rows(j).Font.Color = vbRed  
    End If  
Next j  
  
rng.Cells.Font.Italic = True  
End Sub
```



Per restituire il numero della prima riga in un intervallo, si utilizza la **Proprietà Range.Row** e se l'intervallo specificato contiene più aree, questa proprietà restituirà il numero della prima riga della prima area. Sintassi: *RangeObject.Row*, mentre invece per restituire il numero della prima colonna in un intervallo, si utilizza la **Proprietà Range.Column** e se l'intervallo specificato contiene più aree, questa proprietà restituirà il numero della prima colonna nella prima area. Sintassi: *RangeObject.Column*

Esempi:

Prendi il numero della prima riga nell'intervallo specificato - restituisce 4:

MsgBox ActiveSheet.Range ("B4"). Row

MsgBox Worksheets ("Foglio1"). Range ("B4: D7"). Row

Prendi il numero della prima colonna nell'intervallo specificato - restituisce 2:

MsgBox ActiveSheet.Range ("B4: D7"). Columns

Prendi il numero dell'ultima riga nell'intervallo specificato - restituisce 7:

Spiegazione: *Range ("B4: D7"). Rows.Count* restituisce 4 (il numero di righe nell'intervallo), così come: *Range ("B4:D7"). Rows(Range ("B4:D7"). Rows.Count)* o

Range ("B4:D7"). Rows(4), restituiscono l'ultima riga nell'intervallo specificato.

MsgBox Range ("B4: D7"). Row. (Range ("B4: D7"). Rows.Count). Row

Esempio: Utilizzare la Proprietà Row, Columns per determinare il numero di riga e numero di colonna a righe alternate

```
Sub esempio_2()  
Dim rng As Range, cell As Range, i As Integer  
  
Set rng = Worksheets("Foglio1").Range("B4:D7")  
  
For Each cell In rng  
cell.Value = cell.Row & "," & cell.Column  
Next  
  
For i = 1 To rng.Rows.Count  
If i Mod 2 = 1 Then  
rng.Rows(i).Interior.Color = vbGreen  
Else  
rng.Rows(i).Interior.Color = vbYellow  
End If  
Next  
End Sub
```




È possibile ottenere un riferimento di intervallo in linguaggio VBA utilizzando la **proprietà Range.Address**, che restituisce l'indirizzo di un intervallo come valore stringa. Questa proprietà è di sola lettura. Esempi di utilizzo

MsgBox Range ("B2"). Address 'Restituisce \$B\$2

MsgBox Range ("B2, C3"). Address 'Restituisce \$B\$2, \$C\$3

*Dim strRng As String 'Restituisce \$A\$1: \$B\$2, \$C\$3, \$D\$4
Range ("A1: B2, C3, D4"). Select
strRng = Selection.Address
MsgBox strRng*

MsgBox Range ("B2"). Address (RowAbsolute: = False) 'Restituisce \$B2

MsgBox Range ("B2"). Address (ColumnAbsolute: = False) 'Restituisce B\$2

MsgBox Range ("B2"). Address (ReferenceStyle: = xlR1C1) 'Restituisce R2C2

Restituisce R[1] C[-1] – il Range ("B2") è di 1 riga e -1 colonna rispetto al Range ("C1")

MsgBox Range ("B2"). Address (RowAbsolute: = False, ColumnAbsolute: = False, ReferenceStyle: = xlR1C1, relativeTo: = Range ("C1"))

Restituisce RC [-2] – il Range ("A1") è di 0 riga e -2 colonne rispetto al Range ("C1")

MsgBox Cells(1, 1). Address (RowAbsolute: = False, ColumnAbsolute: = False, ReferenceStyle: = xlR1C1, relativeTo: = Range ("C1"))

■ **Attivare e selezionare Celle con ActiveCell e Selection**

Il metodo **Select** (dell'oggetto Range) viene utilizzato per selezionare una cella o un intervallo di celle in un foglio di lavoro e presenta la seguente Sintassi: *RangeObject.Select*, dovete assicurarvi che il foglio di lavoro in cui viene applicato il metodo Select per selezionare le celle, sia il foglio attivo.

La proprietà **ActiveCell** (dell'oggetto Application) restituisce una singola cella attiva (oggetto Range) nel foglio di lavoro attivo, tenendo presente che la proprietà ActiveCell non funziona se il foglio attivo non è un foglio di lavoro. Quando si seleziona una cella nella finestra attiva, la proprietà **Selection** (dell'oggetto Application) restituisce un oggetto Range che rappresenta tutte le celle che sono attualmente selezionate nel foglio di lavoro attivo.

Una selezione può essere costituita da una singola cella o un intervallo di più celle, ma ci sarà una sola cella attiva al suo interno, che viene restituita utilizzando la proprietà *ActiveCell*. Quando viene selezionata una sola cella, la proprietà ActiveCell restituisce questa cella, mentre invece selezionando più celle utilizzando il metodo *Select*, la prima cella di riferimento diventa la cella attiva, e, successivamente, è possibile modificare la cella attiva utilizzando il metodo *Activate*.

Sia la proprietà ActiveCell e la proprietà Selection sono di sola lettura, e non specificando l'oggetto Application cioè il qualificatore *Application.ActiveCell* o *ActiveCell* o *Application.Selection* o *Selection*, avranno lo stesso effetto. Per attivare una singola cella all'interno della selezione corrente, si deve utilizzare il metodo **Activate** (dell'oggetto Range) che presenta questa Sintassi: *RangeObject.Activate*, e la cella attivata verrà restituita utilizzando la proprietà ActiveCell.



Abbiamo discusso sopra che una selezione può essere costituita da una singola cella o un intervallo di più celle, mentre ci può essere solo una cella attiva all'interno della selezione e quando si attiva una cella al di fuori della selezione corrente, la cella attiva diventa l'unica cella selezionata. È inoltre possibile utilizzare il metodo *Activate* per specificare un intervallo di più celle, ma in effetti sarà attivata solo una singola cella, e questa cella attiva sarà la cella rappresentata dall'angolo superiore sinistro del campo specificato nel metodo. Se questa cella in alto a sinistra si trova all'interno della selezione, la selezione corrente non cambierà, ma se questa cella in alto a sinistra si trova al di fuori della selezione, l'intervallo specificato nel metodo *Activate* diventa la nuova selezione.

Vediamo del codice per illustrare i concetti di *ActiveCell* e *Selection*

Selezione contenente un intervallo di celle e la cella attiva

Range ("C1: F5"). Select 'selezionare il range C1:F5

MsgBox ActiveCell.Address 'restituisce C1, la prima cella di riferimento come cella attiva

Selezione contenente un intervallo di celle e la cella attiva

Range ("F5: C1"). Select 'selezionare il range F5:C1

MsgBox ActiveCell.Address 'restituisce C1, la prima cella di riferimento come cella attiva

Selezione contenente un intervallo di celle e la cella attiva:

Range ("C5: F1"). Select 'selezionare il Range C1: F5

MsgBox ActiveCell.Address 'restituisce C1, la prima cella di riferimento come la cella attiva

Attivare una cella all'interno della selezione corrente

Range ("B6: F10"). Select 'Selezionare il range B6: F10

MsgBox ActiveCell.Address 'restituisce B6, la prima cella di riferimento come la cella attiva

Range ("C8"). Activate 'La selezione è sempre la stessa, ma la cella attiva è ora C8

MsgBox ActiveCell.Address

Attivare una cella al di fuori della selezione corrente

Range ("B6: F10"). Select 'Seleziona l'intervallo B6: F10

MsgBox ActiveCell.Address 'restituisce B6, la prima cella di riferimento, come cella attiva

Range ("A2"). Activate 'la selezione e la cella attiva è ora A2

MsgBox ActiveCell.Address

Selezionare una cella all'interno della selezione corrente

Range ("B6: F10"). Select 'Selezionare l'intervallo B6: F10

MsgBox ActiveCell.Address 'restituisce B6, la prima cella di riferimento come la cella attiva

Range (" C8 "). Select 'sia la selezione che la cella attiva diventano C8

MsgBox ActiveCell.Address

Attivare un intervallo di celle in cui la cella in alto a sinistra si trova all'interno della selezione corrente

Range ("B6: F10"). Select 'Seleziona l'intervallo B6: F10

MsgBox ActiveCell.Address 'restituisce B6, la prima cella di riferimento, come cella attiva

Range ("C8: G12"). Activate 'La selezione rimane la stessa, ma la cella attiva è ora C8

MsgBox ActiveCell.Address

Attivare un intervallo di celle in cui la cella in alto a sinistra si trova al di fuori della selezione corrente

Range ("B6: F10"). Select 'Selezionare l'intervallo B6: F10

MsgBox ActiveCell.Address 'restituisce B6, la prima cella di riferimento, come la cella attiva

Range ("B1: F8"). Activate 'viene cambiato l'intervallo e la selezione, e la cella attiva è ora B1

MsgBox ActiveCell.Address



■ **La proprietà Application.Selection** restituisce l'oggetto selezionato in cui la selezione determina il tipo di oggetto restituito e quando la selezione è un intervallo di celle, questa proprietà restituisce un oggetto *Range*, e questa selezione (che è un oggetto *Range*), può comprendere una singola cella, o più celle o intervalli multipli non contigui. Come detto sopra, il metodo *Select* (dell'oggetto *Range*) è usato per selezionare una cella o un intervallo di celle in un foglio, pertanto, dopo aver selezionato un intervallo, è possibile eseguire azioni sulla selezione di celle utilizzando l'oggetto *Selection*.

```
Sub esempio()  
'Selezionare le celle del foglio attivo utilizzando il metodo Range.Select  
Range("A1:B3,D6").Select  
'impostare il colore di sfondo rosso alle celle della selezione  
Selection.Interior.Color = vbRed  
End Sub
```

■ Le proprietà **Entire Row**, **Entire Column** e **Insert**

La proprietà **Range.EntireRow** restituisce un'intera riga o le righe all'interno del *Range* specificato e restituisce un oggetto *Range* con riferimento alla intera riga. Sintassi: *RangeObject.EntireRow*, mentre invece la proprietà **Range.EntireColumn** restituisce un'intera colonna o le colonne all'interno del *Range* specificato e restituisce un oggetto *Range* con riferimento alla intera colonna.

Sintassi: *RangeObject.EntireColumn*. Esempi di utilizzo delle proprietà *EntireRow* e *EntireColumn*

```
Range ("A2"). EntireRow.Select 'Selezionare la riga 2  
Range ("A2: C4"). EntireRow.Select 'Selezionare le righe 2, 3 e 4  
Cells (3, 4). EntireRow.Cells. (1, 1). Value = 3 'Inserire il valore 3 nel Range A3 esempio  
Range ("A2"). EntireColumn.Select 'Selezionare la colonna A  
Range ("A2: C4"). EntireColumn.Select 'Selezionare le colonne da A a C:  
Cells (3, 4). EntireColumn.Cells (1, 1). Value = 4 'Inserire il valore 4 nel Range D1
```

Il metodo **Range.Insert** si utilizza per inserire una cella o un intervallo di celle in un foglio di lavoro. Sintassi: *RangeObject.Insert (Shift, CopyOrigin)*. Entrambi gli argomenti racchiusi nelle parentesi sono facoltativi.

Quando si inseriscono delle celle, le altre vengono spostate per fare spazio a quelle inserite, ed è possibile impostare un valore per determinare la direzione in cui le altre cellule devono spostarsi.

Specificando *xlShiftDown* si sposteranno le celle in basso e con *xlShiftToRight* si spostano le celle a destra. Tralasciando questo argomento la direzione di spostamento verrà decisa in base alla forma del *Range*.

Specificando *xlFormatFromLeftOrAbove* per l'argomento *CopyOrigin* si copierà il formato delle celle inserite dalle celle sopra a sinistra, e specificando *xlFormatFromRightOrBelow* si copierà il formato dalle celle sotto a destra.

Spostare le celle in basso e copiare la formattazione della cella inserita dalla cella sopra della stessa colonna

Range ("B2"). Insert

Spostare le celle a destra e copiare la formattazione delle celle inserite dalla cella a sinistra

Range ("B2: C4"). Insert

Spostare le celle in basso e copiare la formattazione delle celle inserite a partire dalle celle di inserimento

Range ("B2: D3"). Insert

Spostare le celle in basso e copiare la formattazione delle celle inserite dalle celle in basso

Range ("B2: D3"). Insert CopyOrigin: = xlFormatFromRightOrBelow



Spostare le celle a destra e copiare la formattazione delle celle inserite dalle celle a Range ("B2: D3"). *Insert shift: = xlShiftToRight*

Spostare le celle a destra e copiare la formattazione delle celle inserite dalle celle a destra Range ("B2: D3"). *Insert shift: = xlShiftToRight, CopyOrigin: = xlFormatFromRightOrBelow*

Inserire 2 righe (la n° 2 e 3) e copiare la formattazione delle righe inserite dalle celle di sopra Range ("B2: D3"). *EntireRow.Insert*

Di seguito sono riportati alcuni esempi di inserimento riga o colonna in modo dinamico in un foglio di lavoro.

Esempio: Inserire una riga, n righe sopra l'ultima riga utilizzata.

```
Sub inserisciR2()  
Dim ws As Worksheet, rigaC As Long  
Set ws = Worksheets("Foglio1")  
rigaC = ws.Cells(Rows.Count, "A").End(xlUp).Row 'determinare l'ultima riga utilizzata nella colonna A  
MsgBox rigaC  
'si stabilisce il n° di righe da aggiungere sopra l'ultima riga utilizzata  
n = 5  
'se ci sono abbastanza righe prima dell'ultima riga utilizzato, altrimenti si otterrà un errore  
If rigaC >= n Then  
'se l'ultima riga utilizzata è 5 righe prima dell'inserimento, si inserisce come 1° riga  
ws.Rows(rigaC).Offset(-n + 1, 0).EntireRow.Insert  
Else  
MsgBox "Non ci sono abbastanza righe prima dell'ultima riga utilizzata!"  
End If  
End Sub
```

Esempio: Inserire una riga ogni volta che il valore cercato viene trovato in un intervallo.

```
Sub inserisciR2()  
Dim ws As Worksheet, indirizzo1 As String  
Dim trovaR As Range, cercaR As Range, ultimaC As Range  
Set ws = Worksheets("Foglio1")  
Set cercaR = ws.Range("A1:K100") 'impostare intervallo di ricerca  
MsgBox "Ricerca di 'pippo' nel Range: " & cercaR.Address  
Set ultimaC = cercaR.Cells(cercaR.Cells.Count) 'iniziare la ricerca dopo l'ultima cella nel range  
'trovare il valore specificato, iniziando la ricerca dopo l'ultima cella nel range di ricerca  
Set trovaR = cercaR.Find(What:="pippo", After:=ultimaC, LookIn:=xlValues, lookat:=xlWhole)  
  
If trovaR Is Nothing Then  
MsgBox "Valore non trovato!"  
Exit Sub  
Else  
indirizzo1 = trovaR.Address  
Do  
Set trovaR = cercaR.FindNext(After:=trovaR)  
'quando viene trovato il valore, se il valore viene trovato due volte, verranno inserite 2 righe  
trovaR.Offset(1).EntireRow.Insert  
Loop While trovaR.Address <> indirizzo1  
End If  
End Sub
```



Esempio: Inserire una riga quando un determinato valore viene trovato.

```
Sub inserisciR()  
Dim ws As Worksheet  
Dim cercaRNG As Range, cercaR As Range, ultimaRNG As Range  
Dim ultimaR As Long  
  
Set ws = Worksheets("Foglio1")  
'si deve trovare un valore in questo range e dopo si inserisce una riga  
Set cercaR = ws.Range("A1:E100")  
'si inizia la ricerca dopo l'ultima cella del range di ricerca  
Set ultimaRNG = cercaR.Cells(cercaR.Cells.Count)  
Set cercaRNG = cercaR.Find(What:="pippo", After:=ultimaRNG, LookIn:=xlValues, lookat:=xlWhole)  
If Not cercaRNG Is Nothing Then 'Procedura di uscita se il valore non viene trovato  
    ultimaR = cercaRNG.Row  
    MsgBox ultimaR  
Else  
    MsgBox "Valore non trovato!"  
Exit Sub  
End If  
'se il valore viene trovato (è nella riga 12), si inserisce una riga e passa alla n° 13  
ws.Cells(ultimaR + 1, 1).EntireRow.Insert  
' se il valore rilevato, si inserisce una riga 3 righe di seguito (riga n ° 15)  
ws.Cells(ultimaR + 3, 1).EntireRow.Insert  
'se il valore viene trovato, si inseriscono 3 righe e il valore alla riga 12 sarà spostato alla riga 15  
ws.Cells(ultimaR, 1).Offset(3).EntireRow.Insert  
'oppure  
ws.Cells(ultimaR, 1).EntireRow.Resize(3).Insert  
    ws.Range(Cells(ultimaR, 1), Cells(ultimaR + 2, 1)).EntireRow.Insert  
    ws.Rows(ultimaR & ":" & ultimaR + 2).EntireRow.Insert Shift:=xlDown  
  
    ws.Cells(ultimaR + 1, 1).EntireRow.Resize(3).Insert  
    ws.Range(ws.Cells(ultimaR + 1, 1), ws.Cells(ultimaR + 3, 1)).EntireRow.Insert  
  
ws.Cells(ultimaR + 2, 1).EntireRow.Resize(3).Insert  
End Sub
```

Esempio: Inserire una riga o colonna, specificando la riga/colonna da inserire

```
Sub inserireRC()  
    Dim ws As Worksheet  
    Set ws = Worksheets("Foglio1")  
'specificare il numero di righe da inserire  
ws.Rows(12).Insert  
'specificare il range sotto al quale inserire le righe  
ws.Range("C3").EntireRow.Offset(1, 0).Insert  
'specificare il numero di colonne da inserire  
ws.Columns(4).Insert  
'specificare l'intervallo a destra del quale inserire le colonne  
ws.Range("C3").EntireColumn.Offset(0, 1).Insert  
End Sub
```




Esempio: Inserisci righe (numero definito dall'utente) all'interno di valori consecutivi che si trovano in una colonna

```
Sub inserisciR3()  
Dim ws As Worksheet, rng As Range  
Dim ultimaRU As Long, rigaI As Long, Rcella As Long, Ccella As Long  
  
Set ws = Worksheets("Foglio1")  
ws.Activate  
'impostare il numero di colonna in cui due valori consecutivi sono controllati per inserire le righe  
Ccella = 1  
'impostare il numero di riga da dove iniziare la ricerca dei valori consecutivi  
Rcella = 1  
'determinare l'ultima riga utilizzata nella colonna  
ultimaRU = Cells(Rows.Count, Ccella).End(xlUp).Row  
'inserire il numero di righe da inserire tra due valori consecutivi  
rigaI = InputBox("Inserisci il numero di righe da inserire")  
  
If rigaI < 1 Then  
MsgBox "Errore - inserire un valore uguale o superiore a 1"  
Exit Sub  
End If  
  
MsgBox "Questo codice inserirà " & rigaI & " righe, ovunque si trovano valori consecutivi nella  
colonna numero " & Ccella & ", iniziando la ricerca dalla riga numero " & Rcella  
'Loop fino al numero di riga corrisponde l'ultima riga utilizzata  
Do While Rcella < ultimaRU  
Set rng = Cells(Rcella, Ccella)  
'Nel caso di due valori consecutivi  
If rng <> "" And rng.Offset(1, 0) <> "" Then  
'Inserire il numero di righe definito dall'utente  
Range(rng.Offset(1, 0), rng.Offset(rigaI, 0)).EntireRow.Insert  
Rcella = Rcella + rigaI + 1  
'Determinare l'ultima riga utilizzata in modo dinamico e cambia l'inserimento delle righe  
ultimaRU = Cells(Rows.Count, Ccella).End(xlUp).Row  
'Metodo alternativo per determinare l'ultima riga utilizzata  
Else  
Rcella = Rcella + 1  
End If  
Loop  
  
End Sub
```