

**FAG** \*pro  
DigitalLifeStyle

# **Excel: macro e VBA**

**Alessandra Salvaggio**

EDIZIONI  
**FAG**  
MILANO

Excel: macro e VBA

**Autore:** Alessandra Salvaggio

**Collana:**

 \*pro  
DigitalLifeStyle

**Progetto editoriale:** Fabrizio Comolli

**Progetto grafico:** escom - Milano

Prima edizione

**ISBN:** 88-8233-482-1

Copyright © 2005 **Edizioni FAG Milano**

Via G. Garibaldi 5 – 20090 Assago (MI) - [www.fag.it](http://www.fag.it)

Finito di stampare: nel mese di ottobre 2005 da escom - Milano

*Nessuna parte del presente libro può essere riprodotta, memorizzata in un sistema che ne permetta l'elaborazione, né trasmessa in qualsivoglia forma e con qualsivoglia mezzo elettronico o meccanico, né può essere fotocopiata, riprodotta o registrata altrimenti, senza previo consenso scritto dell'editore, tranne nel caso di brevi citazioni contenute in articoli di critica o recensioni.*

*La presente pubblicazione contiene le opinioni dell'autore e ha lo scopo di fornire informazioni precise e accurate. L'elaborazione dei testi, anche se curata con scrupolosa attenzione, non può comportare specifiche responsabilità in capo all'autore e/o all'editore per eventuali errori o inesattezze.*

*Nomi e marchi citati nel testo sono generalmente depositati o registrati dalle rispettive aziende. L'autore detiene i diritti per tutte le fotografie, i testi e le illustrazioni che compongono questo libro.*



## 5. Il controllo calendario

In questo capitolo inizieremo a occuparci di un nuovo progetto per imparare a **lavorare con le date**. Come prima operazione, inseriremo una data nel foglio di lavoro attraverso un **controllo calendario**. Trovate l'esempio svolto sul booksite nel file `calendario.xls`, nel foglio `O1` nel form `frmO1Calendario`.

Preparate un foglio Excel come quello mostrato nella Figura 5.1.

Il pulsante di comando si chiama `cmdInserisciData`. Ora visualizzate il codice relativo a questo foglio e fate in modo che, alla pressione di tale pulsante, si apra un form chiamato `frmCalendario`. Create anche questo form e assegnategli il nome corretto (sono tutte operazioni che avete imparato nei capitoli precedenti).

Visualizzate l'oggetto form che avete realizzato. Dobbiamo inserire sul form un calendario dal quale scegliere la data del prestito da inserire nella colonna C.

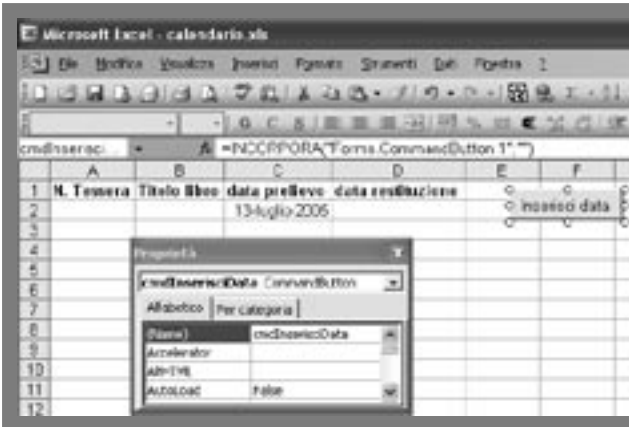


Figura 5.1 - Il foglio Excel pronto per l'esercizio.

Ci occorre il controllo calendario, per averlo a disposizione nella casella degli strumenti dell'editor di VBA selezionate **Strumenti > Controlli aggiuntivi** e mettete un segno di spunta accanto a **Controllo Calendario 11.0**. Premete **Ok** per chiudere la finestra **Controlli aggiuntivi**.

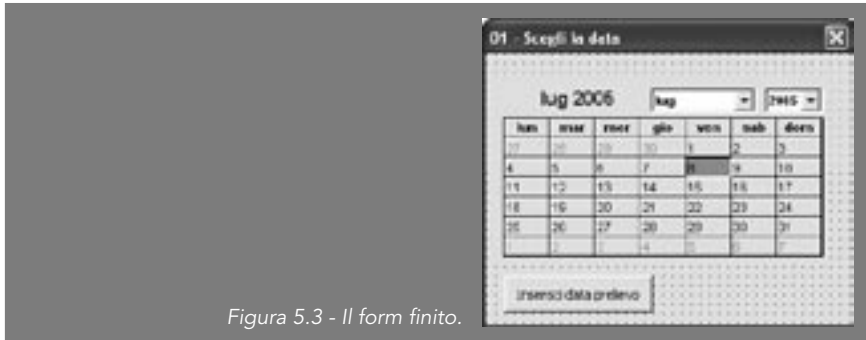
Se utilizzate una versione diversa di Office il controllo calendario potrebbe avere un'altra numerazione, ma funzionerà comunque bene.

Ora sulla casella degli strumenti dovreste vedere il nuovo controllo calendario (Figura 5.2).



Figura 5.2 - Il controllo calendario sulla casella degli strumenti

Fate clic sul controllo calendario e disegnate un calendario sul form sufficientemente grande per mostrare chiaramente tutte le date di un mese. Notate che il calendario si chiama Calendar, lasciamo pure questo nome. Aggiungete al form un pulsante di comando con nome cmdInserisciPrelievo e caption Inserisci data prelievo. Ora il vostro form dovrebbe presentarsi come mostrato nella Figura 5.3.



Fate un doppio clic sul pulsante appena inserito per visualizzare il codice relativo al form con la routine per l'evento clic sul pulsante. Completate la routine come segue:

```
Private Sub cmdInserisciPrelievo_Click()
    ActiveCell = Calendar1.Value
    Selection.NumberFormat = "d-mmmm-yyyy"
    Me.Hide
End Sub
```

La prima riga di questo codice `ActiveCell = Calendar1.Value` assegna alla cella attiva il valore del controllo calendario, ossia la data selezionata dall'utente (se avessi scritto `Range("A1") = Calendar1.Value`, avrei inserito la data corrente nella cella A1).

La seconda riga `Selection.NumberFormat = "d-mmmm-yyyy"` utilizza la proprietà **NumberFormat** dell'oggetto **Selection** (che rappresenta le celle selezionate nel foglio) per formattare la data inserita.

La cella attiva è anche selezionata quindi posso indicarla attraverso l'oggetto Selection, se avessi scritto la data utilizzando l'oggetto Range, prima di formattare la data con l'istruzione Selection.NumberFormat = "d-mmmm-yyyy", avrei dovuto selezionare la cella in cui ho scritto con l'istruzione Range(nome\_della\_cell).Select.

Il formato è descritto utilizzando la lettera d per rappresentare il giorno, m per il mese e y per l'anno. La ripetizione delle lettere indica il numero di cifre da utilizzare per esprimere ciascun valore. Se la lettera che rappresenta il giorno o il mese viene ripetuta quattro volte, il giorno e il mese vengono espressi in parola. Nel nostro caso il giorno è espresso con due cifre, il mese in parola e l'anno con quattro cifre. Se avessi scritto "dddd, dd-mm-yyyy", il giorno sarebbe stato espresso dal giorno della settimana scritto per intero in parola (per esempio lunedì) e da un numero a due cifre, il mese con un numero a due cifre e l'anno con quattro (per esempio: sabato, 11-06-05).

Per fare in modo che all'apertura del form il controllo calendario mostri la data corrente (e non l'ultima data inserita), occorre aggiungere al codice già scritto la routine che segue:

```
Private Sub UserForm_Activate()  
    Calendar1.Value = Date  
End Sub
```

Già conoscete l'evento Activate dell'oggetto form, concentriamoci quindi sull'istruzione Calendar1.Value = Date.

**Date** è una funzione, ossia un insieme di operazioni a cui viene assegnato un nome. Richiamando quel nome vengono eseguite le operazioni memorizzate. È un po' come se si trattasse di una ricetta di cucina, ognuna ha un nome, per esempio "pasta alla carbonara" ed è costituita da una serie di operazioni, per esempio "prendere del guanciale, tagliarlo a pezzetti ecc.". A chi conosce la ricetta non devo indicare tutta la sequenza di operazioni, basta dire "cucina la pasta alla carbonara".

VBA conosce la "ricetta" `Date` e sa che, quando gli viene richiesta, deve leggere e restituire la data del computer.

Con l'istruzione `Calendar1.Value = Date` facciamo sì che il valore della data del computer sia assegnato al controllo calendario, che mostrerà la data corrente quando viene attivato il form che lo contiene.

Una funzione restituisce sempre un valore. In questo caso il valore è la data corrente.

## Conclusioni

In questo capitolo abbiamo imparato a:

- aggiungere alla casella degli strumenti il controllo calendario;
- inserire sul form un controllo calendario;
- inserire sulla cella attiva del foglio di lavoro la data ricavata dal controllo calendario;
- formattare la data inserita con la proprietà **FormatNumber**;
- utilizzare la funzione **Date**;
- mostrare nel controllo calendario la data corrente.



## 6. Calcolare i giorni lavorativi

In questo capitolo continueremo a occuparci di date. In particolare, impareremo a utilizzare, all'interno di VBA, una funzione del foglio di lavoro Excel per calcolare **una data che ricorre** dopo un determinato numero di **giorni lavorativi**. Trovate l'esempio svolto sul booksite nel file `calendario.xls`, nel foglio `02` nel form `frm02Calendario`.

Continueremo a lavorare sul file del capitolo precedente: inseriremo un pulsante che permette di inserire nella colonna accanto a quella in cui si è inserita la data di prelievo del libro, la data in cui il libro deve essere restituito, calcolata aggiungendo alla prima data 15 giorni lavorativi (ossia escludendo i sabati, le domeniche e le festività).

Come prima operazione, inserite un nuovo foglio di lavoro Excel al file e chiamatelo `Vacanze`.



Nella colonna A del nuovo foglio inserite le date da considerare festive nell'anno in corso (Figura 6.1).

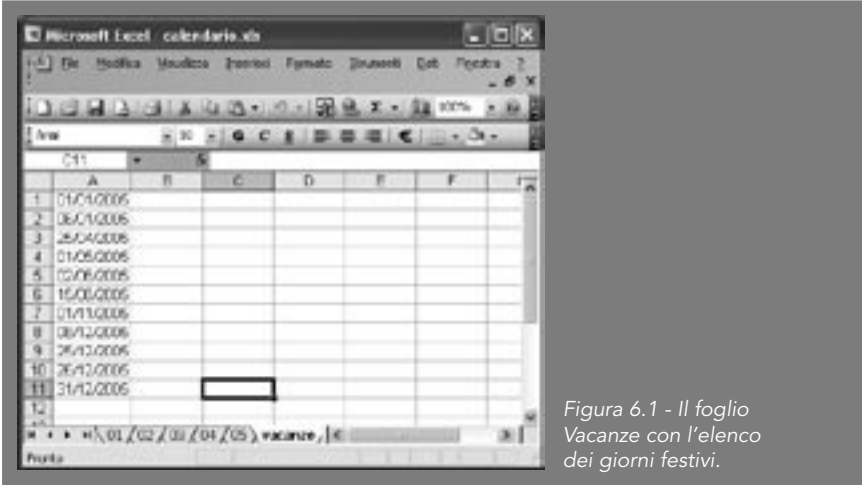


Figura 6.1 - Il foglio Vacanze con l'elenco dei giorni festivi.

Occorre assegnare alle celle che contengono i giorni festivi (nel nostro esempio A1:A11) un nome che ci permetterà di richiamarle in blocco all'interno del codice. Scegliamo il nome **Vacanze**.

Per assegnare un nome a un intervallo di celle, selezionatele, quindi scegliete **Inserisci > Nome > Definisci**. Excel vi mostrerà la finestra **Definisci nome**. Nella casella **Nomi nella cartella di lavoro** scrivete il nome che volete utilizzare (nel nostro esempio Vacanze), quindi premete il pulsante **OK**.

Ora tornate all'editor di VBA e riaprite il form frmCalendario e aggiungete un pulsante di comando con nome cmdDataRestituzione e caption Inserisci data restituzione.

Fate doppio clic sul nuovo pulsante per visualizzare il codice relativo al foglio. Completate la routine cmdDataRestituzione\_Click() come segue:

```
Private Sub cmdDataRestituzione_Click()
    ActiveCell.Offset(0, 1).Activate
```

```
ActiveCell.FormulaR1C1= "=GIORNO.LAVORATIVO(R[0]C[-1],15,vacanze)"
Selection.NumberFormat = "d-mmm-yyy"
End Sub
```

Dobbiamo discutere le prime due righe all'interno della routine. Conoscete già il significato della terza.

La proprietà **Offset** di un oggetto Cella (in questo **ActiveCell**, ossia la cella attiva) permette di spostarsi dalla cella (o intervallo) attivo di un numero di righe e colonne pari a quelle specificate come **argomenti** della proprietà stessa. Si chiama **argomento** ogni valore inserito fra le parentesi tonde poste dopo una proprietà, un metodo o una funzione. Questi valori permettono di definire il comportamento della proprietà, del metodo o della funzione.

In questo caso diciamo a Excel di spostarsi di 0 righe e di una colonna a sinistra, infatti il primo argomento della proprietà Offset indica di quante righe effettuare lo spostamento, mentre il secondo indica il numero delle colonne. Per quanto riguarda le righe, un numero positivo porta a uno spostamento verso il basso, un numero negativo porta a uno spostamento verso l'alto. Per quanto riguarda le colonne, a un numero positivo corrisponde uno spostamento verso destra, mentre a un numero negativo verso sinistra.

Una volta scelta la cella con Offset, usiamo il metodo **Activate** per attivarla e impostarla come cella attiva.

Dopo aver attivato la cella in cui vogliamo inserire la data di restituzione del libro, utilizziamo la proprietà **FormulaR1C1** per specificare quale funzione del foglio Excel (quindi non una funzione VBA) vogliamo scrivere in quella cella. R1C1 indica che nella funzione si usano riferimenti in cui sia le righe sia le colonne del foglio di lavoro sono numerate. Nello stile R1C1 la posizione di una cella viene indicata da una R seguita da un numero di riga e da una C seguita da un numero di colonna. Se, come nel nostro esempio, i numeri di riga e colonna sono posti fra parentesi quadre, devono intendersi come relativi alla cella attiva.

Analizziamo la funzione "=GIORNO.LAVORATIVO(R[0]C[-1],15,vacanze)" che stiamo usando. La funzione GIORNO.LAVORATIVO restituisce una data ottenuta sommando alla data specificata come primo argomento (in questo caso R[0]C[-1], ossia la data contenuta nella cella della colonna di sinistra rispetto a quella attiva, quindi la data di prestito) il numero di giorni lavorativi indicati come secondo argomento (in questo

caso 15). Si tiene conto, oltre che dei sabati e delle domeniche, anche dei giorni festivi specificati nel terzo argomento (in questo caso tutte le date comprese nell'intervallo a cui abbiamo assegnato il nome Vacanze).

I valori dell'intervallo Vacanze dovranno essere aggiornati ogni anno. Dopo l'aggiornamento delle date dovrete ricreare l'intervallo denominato Vacanze, affinché questo comprenda sempre tutte le celle che contengono i giorni festivi dell'anno.

Prima di passare a Excel e provare il codice, bisogna assicurarsi che la funzione `GIORNO.LAVORATIVO` sia disponibile nel foglio di lavoro, infatti non si tratta di una funzione base, ma va aggiunta a Excel.

Per far questo, tornate al foglio di lavoro (quindi non all'interno dell'editor di VBA) e scegliete **Strumenti > Componenti aggiuntivi**. Excel vi mostrerà la finestra **Componenti aggiuntivi**. Mettete una spunta all'opzione **Strumenti di Analisi** (Figura 6.2), quindi premete OK per chiudere la finestra.



Figura 6.2 - Aggiungere a Excel le funzioni di Analisi.

Ora potete fare clic in una cella della colonna C e provate il codice. Notate che dove visualizzate la data di restituzione, in realtà è stata inserita la funzione che calcola la data (Figura 6.3) e non la data stessa.

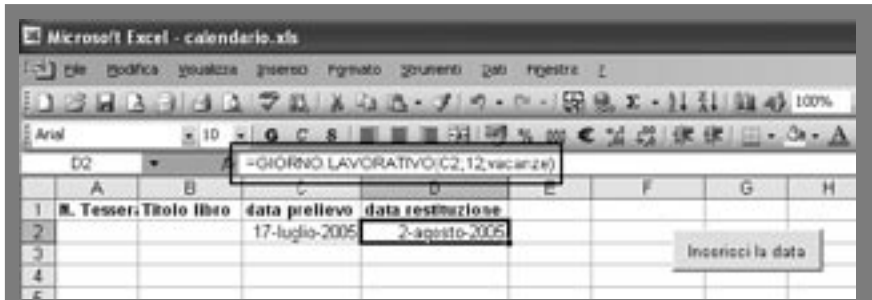


Figura 6.3 - Aggiungere a Excel le funzioni di Analisi.

## Conclusioni

In questo capitolo abbiamo imparato a:

- utilizzare la proprietà **Offset** per selezionare celle poste a una data distanza da quella o quelle attive;
- attivare una cella col metodo **Activate**;
- usare la proprietà **FormulaR1C1** per impiegare all'interno di VBA le funzioni del foglio di lavoro Excel;
- calcolare una data che ricorre dopo un determinato numero di giorni lavorativi;
- aggiungere nuove funzioni al foglio di lavoro.