

Excel VBA Masterclass

INTRODUZIONE AL VISUAL BASIC EDITOR (VBE)



Excel Academy

Introduzione al Visual Basic Editor (VBE)

In questo capitolo, inizierai ad imparare come utilizzare il Visual Basic Editor, un potente strumento in dotazione con Office per lavorare con VBA. Questo editor di programmazione rappresenta il culmine di oltre 20 anni di modifiche e miglioramenti. È molto efficace.

Tutte le applicazioni che ospitano VBA utilizzano il Visual Basic Editor; quindi, l'interfaccia e gli strumenti sono praticamente gli stessi indipendentemente dall'applicazione utilizzata. Questo capitolo illustra i fondamenti del Visual Basic Editor: i suoi componenti, cosa fanno e come li usi. Imparerai le manovre più avanzate lavorando con VBA più avanti in questo corso.

Sommario

Introduzione al Visual Basic Editor (VBE)	1
Aprire il Visual Basic Editor	3
Aprire il Visual Basic Editor con una macro selezionata	3
Aprire direttamente il Visual Basic Editor	4
Raggiungere una macro	5
Le finestre principali del Visual Basic Editor	7
La finestra Gestione progetti	8
Il Visualizzatore oggetti	14
La finestra Codice	15
La finestra Proprietà	22
La finestra Immediata	26
Impostazione delle proprietà di un progetto	27

Aprire il Visual Basic Editor

Il Visual Basic Editor si apre dall'applicazione Office che stai utilizzando. Ad esempio, se lavori in Excel, apri Visual Basic Editor da Excel. L'istanza di Visual Basic Editor che apri viene quindi associata a Excel. Tuttavia, è possibile aprire più istanze di Visual Basic Editor. Ad esempio, se l'hai già aperta in Excel, è possibile aprirne un'altra istanza in Word e poi un'altra in Access. È possibile aprire Visual Basic Editor in due modi:

- Selezionando una macro che vuoi modificare. Excel apre quindi il Visual Basic Editor e visualizza quella macro in modo che tu sia pronto a lavorarci
- Aprire direttamente l'Editor e quindi individuare il codice macro con cui si desidera lavorare

Le prossime due sezioni illustreranno i due modi di aprire Visual Basic Editor e la terza sezione mostrerà come raggiungere una macro.

Aprire il Visual Basic Editor con una macro selezionata

Se conosci il nome della macro su cui desideri lavorare, utilizza questo metodo per aprire il Visual Basic Editor e visualizzare immediatamente la macro. In questo esempio apriremo la macro `Aggiungi_Mesi` che hai registrato in una delle sezioni precedenti:

1. Apri Excel se non è già in esecuzione
2. Premi `Alt + F8` per visualizzare la finestra di dialogo Macro
3. Seleziona la macro `Aggiungi_Mesi` e fai clic sul pulsante Modifica.
Excel apre il Visual Basic Editor con la macro visualizzata e pronta per la modifica, come mostrato in **Figura 1**.

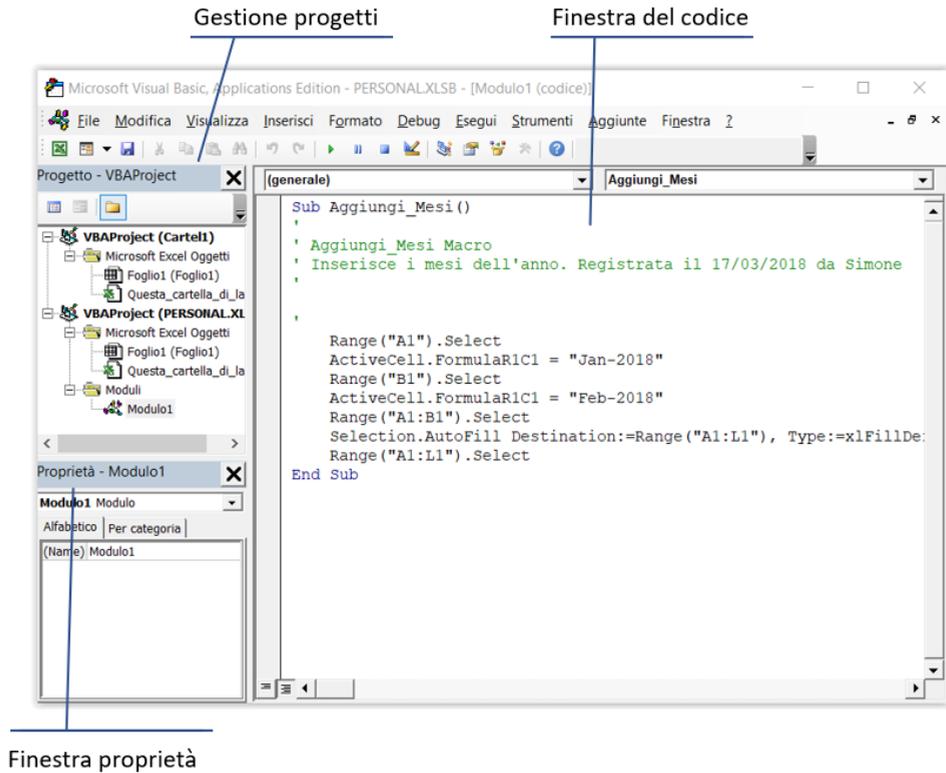


Figura 1 - Il Visual Basic Editor con la macro *Aggiungi_Mesi* si apre nella finestra del codice

4. Seleziona File ➤ Chiudi per chiudere il Visual Basic Editor per il momento in modo da poterlo aprire usando il metodo descritto nella prossima sezione

Aprire direttamente il Visual Basic Editor

Per aprire direttamente il Visual Basic Editor, attieniti alla seguente procedura:

1. Apri o attiva Excel
2. Premi Alt + F11. Si apre l'editor di Visual Basic

Se non vedi la Finestra Proprietà (vedi **Figura 1**), premi F4. Imparerai di più su questa importante finestra a breve.

L'editor di Visual Basic ricorda le finestre del codice

A seconda dello stato del Visual Basic Editor l'ultima volta che è stato chiuso, è possibile visualizzare una o più finestre di codice aperte. Ad esempio, se hai lasciato la Finestra del codice del modulo NuoveMacro aperta nella sessione di lavoro precedente, il VBE visualizzerà nuovamente questa finestra del codice.

Raggiungere una macro

Dopo aver aperto direttamente il Visual Basic Editor, utilizza il riquadro “Gestione progetti” (mostrato nella parte sinistra in **Figura 1**) per accedere alla macro. Il riquadro “Gestione progetti”, si utilizza inoltre per spostarsi tra progetti e moduli aperti quando si lavora nel Visual Basic Editor.

Gestione progetti è simile a Esplora file di Windows

Il riquadro “Gestione progetti” funziona come la struttura ad albero standard del File Explorer di Windows quando stai visualizzando cartelle e sottocartelle. Potresti vedere diversi progetti visualizzati nell'albero (ne parleremo più avanti nel capitolo).

Per accedere alla macro `Aggiungi_Mesi`, attieniti alla seguente procedura:

1. Nel riquadro Gestione progetti, nell'angolo superiore sinistro del Visual Basic Editor, espandi la voce “VBAProject (PERSONAL.XLSB)” facendo clic sul segno + a sinistra del nome (se la voce è già espansa, salta questo passaggio)
2. Fai doppio clic sulla voce Moduli per espanderla
3. Fai doppio clic sul modulo Modulo1. Il Visual Basic Editor visualizza il contenuto del modulo nella finestra Codice sul lato destro, come puoi vedere in **Figura 1**.

Se il modulo contiene più di una macro, dovrai anche selezionare la macro con cui desideri lavorare, in questo caso, la macro `Aggiungi_Mesi` (se hai registrato solo la macro `Aggiungi_Mesi`, solo questa macro viene visualizzata nella finestra del codice).

Per selezionare una macro, utilizza uno di questi metodi:

- Nella finestra Codice, seleziona la macro dall'elenco a discesa Routine, come mostrato in **Figura 2** (se si posiziona il puntatore del mouse sopra il menu a tendina prima di aprirlo, verrà visualizzato un *tooltip* con il suo nome: Routine)
- Usa la barra di scorrimento per scorrere fino alla macro che vuoi modificare, che è identificata dalla parola Sub, il nome che hai dato e una coppia di parentesi, in questo caso, `Sub Aggiungi_Mesi()`.

Massimizza la finestra del codice

Per impostazione predefinita, la finestra del codice viene visualizzata in una dimensione della finestra "normale". In altre parole, c'è uno sfondo grigio attorno ad essa e non riempie tutta l'area. Questo ti consente di aprire altre finestre codice nella stessa area. Tuttavia, per me a volte è po' scomodo, quindi tendo a visualizzare sempre la finestra Codice ingrandita, come mostrato in **Figura 1** o in **Figura 2**. Questo rende più facile vedere il codice. Per

visualizzare solo una finestra Codice ingrandita, fai clic sul pulsante Ingrandisci della finestra Codice, a sinistra del pulsante X rosso che chiude la finestra.

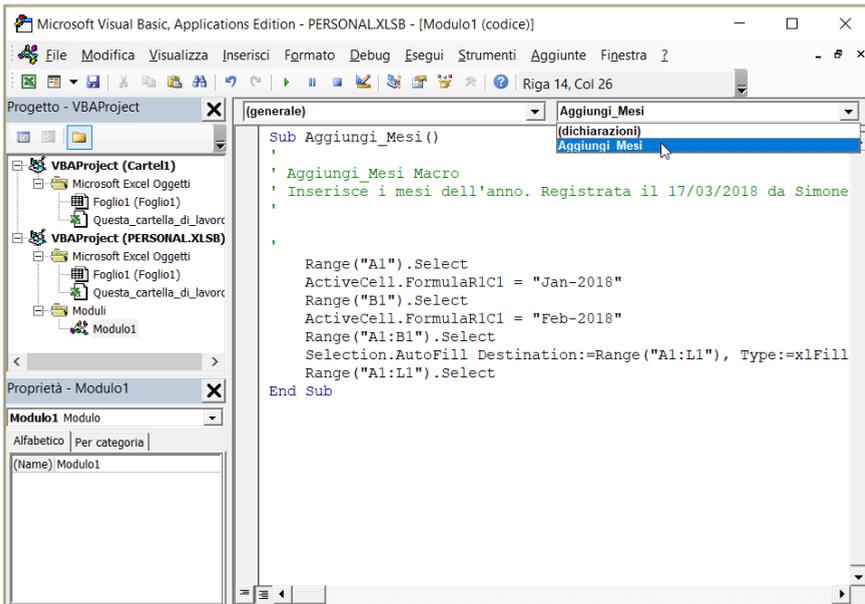


Figura 2 - Se il modulo contiene due o più macro, scorri fino alla macro che vuoi modificare o selezionala dall'elenco a discesa Routine.

Le finestre principali del Visual Basic Editor

Nelle sezioni seguenti, imparerai come utilizzare le finestre principali del Visual Basic Editor.

La finestra Gestione progetti

Gestione progetti è lo strumento da usare per la navigazione tra i vari oggetti nel Visual Basic Editor. La **Figura 3** mostra la finestra Gestione progetti per Excel. Ogni progetto può contenere alcuni o tutti i seguenti elementi (ma non preoccuparti ora di oggetti come moduli di classe, librerie di link e così via: li esploreremo nei prossimi capitoli).

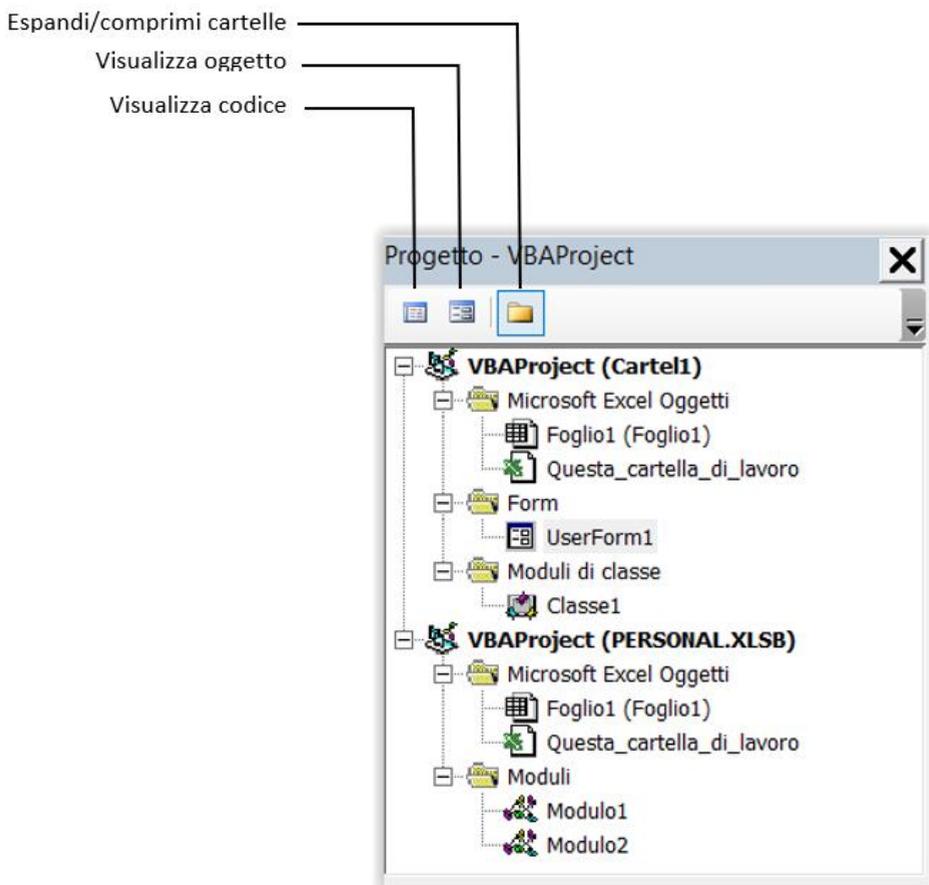


Figura 3 - Utilizza Gestione progetti per accedere al modulo con cui vuoi lavorare.

- UserForms (finestre che fanno parte dell'interfaccia utente della macro, ad esempio una finestra di dialogo personalizzata che accetta l'input dell'utente)
- Moduli contenenti macro, procedure e funzioni
- Moduli di classe (moduli specializzati che definiscono gli oggetti, le loro proprietà e i loro valori)
- Riferimenti ad altri progetti o ai file di libreria (come DLL-Dynamic Link Libraries)
- Oggetti relativi all'applicazione

Ad esempio, ogni cartella di lavoro di Excel contiene un oggetto di classe denominato **Questa_cartella_di_lavoro** che consente di accedere alle proprietà e agli eventi della cartella di lavoro, e un oggetto Foglio (denominato Foglio1, Foglio2 e così via) per ciascun foglio di lavoro.

Ogni documento o modello aperto è considerato un progetto separato e viene visualizzato come radice nella struttura del progetto. La struttura del progetto contiene anche eventuali contenitori di archiviazione macro globali, ad esempio la cartella di lavoro macro personale in Excel (PERSONAL.XLSB), e tutti i componenti aggiuntivi caricati.

Cambia il nome di un progetto in qualsiasi momento

È possibile modificare il nome di un progetto utilizzando la finestra di dialogo Proprietà progetto (discussa più avanti in questo capitolo) o selezionando il progetto e immettendo un nuovo nome nel riquadro Proprietà, mostrato direttamente sotto il riquadro Gestione progetti.

Una volta modificato il nome, il progetto viene identificato con quel nome in Gestione progetti, seguito dal nome del documento o del modello.

Per esplorare Gestione progetti, occorre fare clic sul segno più (riquadrato) a sinistra di un elemento del progetto in modo da espandere la vista e visualizzare gli elementi contenuti nel progetto, quindi facendo clic sul segno meno risultante per comprimere la vista e nascondere di nuovo gli elementi. È possibile doppio clic su un modulo per visualizzarne il codice nella finestra del codice. Nello stesso modo è possibile visualizzare una UserForm.

Visual Basic Editor visualizza Gestione progetti per impostazione predefinita e, poiché Gestione progetti offre una navigazione rapida ed efficiente tra i vari elementi dei progetti VBA, in genere è più semplice lasciarlo visualizzato a meno che tu non sia a corto di spazio sullo schermo oppure se devi lavorare per lunghi periodi nella finestra del codice e non hai bisogno di passare ad altri elementi.

Per chiudere Gestione progetti, fai clic sul suo pulsante di chiusura (il pulsante X nella barra del titolo). Per visualizzare nuovamente Gestione progetti, premi **Ctrl + R** o seleziona **Visualizza > Gestione progetti**. Come vedrai più avanti in questo capitolo, puoi anche sganciare la finestra Gestione progetti. Questo ti consente di metterla da parte quando hai bisogno di più spazio. Ma non occupa molto spazio, quindi, di nuovo, molte persone lo lasciano semplicemente in alto a sinistra.

Nella **Figura 3**, appaiono tre pulsanti nella parte superiore di Gestione progetti:

Visualizza codice: mostra la finestra Codice per l'oggetto selezionato. Ad esempio, se si seleziona una UserForm in Gestione progetti e si fa clic sul pulsante Visualizza codice, Visual Basic Editor visualizza una finestra Codice contenente qualsiasi codice collegato alla UserForm. Se si seleziona un modulo o un modulo di classe in Gestione progetti e si fa clic sul pulsante Visualizza codice, Visual Basic Editor visualizza una finestra di codice contenente il codice presente nel modulo. Puoi anche fare clic con il pulsante destro del mouse su un elemento in Gestione progetti e scegliere Visualizza codice dal menu di scelta rapida.

Codice è semplicemente un sinonimo di programmazione: la serie di comandi che inserisci (o registri) per far sì che il computer si comporti in un certo modo. Il codice viene talvolta chiamato codice di programmazione o codice sorgente.

Si noti che le parole utilizzate in termini di programmazione come **Selection** o **End Sub** utilizzati da un linguaggio di programmazione per computer come VBA sono indicate da una varietà di sinonimi: istruzioni, parole chiave, comandi e così via.

Doppio clic su un modulo per vedere il suo codice

Se hai un modulo o un modulo di classe, puoi anche fare doppio clic sull'oggetto per visualizzarne il codice. Di solito è più veloce che selezionarlo e quindi fare clic sul pulsante *Visualizza codice*. Per una UserForm o un file, tuttavia, facendo doppio clic viene visualizzata l'opzione *Visualizza oggetto* (discussa di seguito) anziché l'opzione *Visualizza codice*.

Visualizza oggetto: visualizza una finestra contenente l'oggetto selezionato.

Il pulsante *Visualizza oggetto* rimane inattivo e disattivato finché non si seleziona un oggetto (come una UserForm o un file o un oggetto all'interno di un file) che possa essere visualizzato. Se l'oggetto selezionato è una UserForm, facendo clic sul pulsante *Visualizza oggetto* viene visualizzata la UserForm; se l'oggetto selezionato è un file o un oggetto all'interno di un file, facendo clic sul pulsante *Visualizza oggetto* viene visualizzato l'oggetto nella finestra di Excel.

Ad esempio, selezionando un oggetto *Foglio1* in una cartella di lavoro di Excel e facendo clic sul pulsante *Visualizza oggetto* viene visualizzato il foglio di lavoro nella cartella di lavoro di Excel, all'interno della finestra di Excel.

Visualizzare un oggetto

È inoltre possibile attivare la modalità Visualizza oggetto facendo clic con il pulsante destro del mouse su un oggetto e scegliendo **Visualizza oggetto** dal menu di scelta rapida o facendo doppio clic su un oggetto che supporta la funzione Visualizza oggetto (se l'oggetto non supporta la funzione Visualizza oggetto, facendo doppio clic su di esso si attiva invece la modalità Visualizza codice).

Espandi/comprimi cartelle: espande o comprime la vista degli oggetti in Gestione progetti tra la visualizzazione delle cartelle (una vista che mostra gli oggetti raggruppati all'interno dei progetti e delle cartelle) e la visualizzazione dei contenuti (che visualizza solo gli oggetti all'interno dei loro progetti). La parte sinistra della **Figura 4** mostra Gestione progetti che visualizza cartelle e la parte destra mostra Gestione progetti nella visualizzazione contenuto. Se trascorrerai più tempo nella visualizzazione cartelle o nella visualizzazione contenuti dipenderà dalle dimensioni dello schermo, dal numero di oggetti inseriti in un dato progetto e dal modo in cui la tua mente lavora, non necessariamente in questo ordine. Per molti scopi, ti consigliamo di alternare la visualizzazione di cartelle e la visualizzazione di contenuti per localizzare gli oggetti più facilmente.

Gestione progetti ha diversi usi, perciò è una buona ragione per tenerlo sempre aperto.

Oltre alla navigazione verso gli oggetti con cui devi lavorare, con Gestione progetti puoi eseguire le seguenti attività aggiuntive:

- Aggiungere o rimuovere componenti da un progetto. Ad esempio, è possibile utilizzare Gestione progetti per aggiungere un modulo o una UserForm a un progetto

- Confrontare i componenti di un progetto con i componenti di un altro progetto. Un confronto come questo può essere utile quando è necessario stabilire rapidamente le differenze tra due o più progetti (ad esempio, la copia di backup di un modello aziendale e le copie a cui gli utenti hanno lavorato).

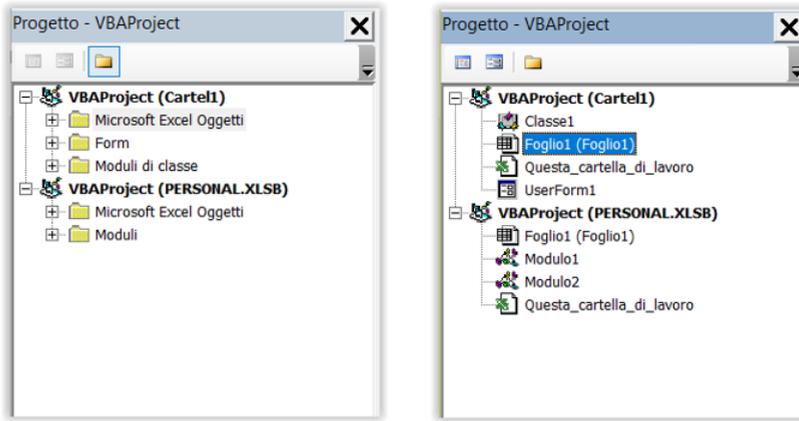


Figura 4 - La vista Cartelle (a sinistra) visualizza gli oggetti separati in cartelle al di sotto dei progetti che li contengono. La vista contenuti (a destra) visualizza solo gli oggetti e i progetti che li contengono.

- Spostare o copiare elementi da un progetto all'altro. È possibile trascinare un modulo di codice, un modulo di classe o una UserForm da un progetto a un altro in Gestione progetti per copiarlo, oppure anche tra due istanze di Visual Basic Editor.
- Importare o esportare un modulo di codice o una UserForm da o verso un progetto.

Gestione progetti è la tua miglior scelta

Molte azioni che è possibile eseguire tramite Gestione progetti possono essere eseguite anche tramite le voci di menu dell'Editor di Visual Basic. In generale, però, Gestione progetti offre il modo più semplice per navigare nel Visual

Basic Editor, soprattutto se si hanno contemporaneamente diversi progetti complessi aperti. È possibile accedere alle funzioni più utilizzate per un oggetto facendo clic con il pulsante destro del mouse su Gestione progetti, visualizzando così il menu di scelta rapida.

Il Visualizzatore oggetti

Visual Basic Editor fornisce un Visualizzatore oggetti completo per lavorare con gli oggetti in VBA. Guarderemo il Visualizzatore oggetti in dettaglio più avanti. Nel frattempo, dai un'occhiata alla **Figura 5**, che mostra il Visualizzatore oggetti in Excel. L'oggetto **AutoFilter** è selezionato nel pannello di sinistra e un elenco delle sue proprietà viene visualizzato nel pannello a destra (per vedere la stessa cosa nel tuo editor VBA, premi F2).

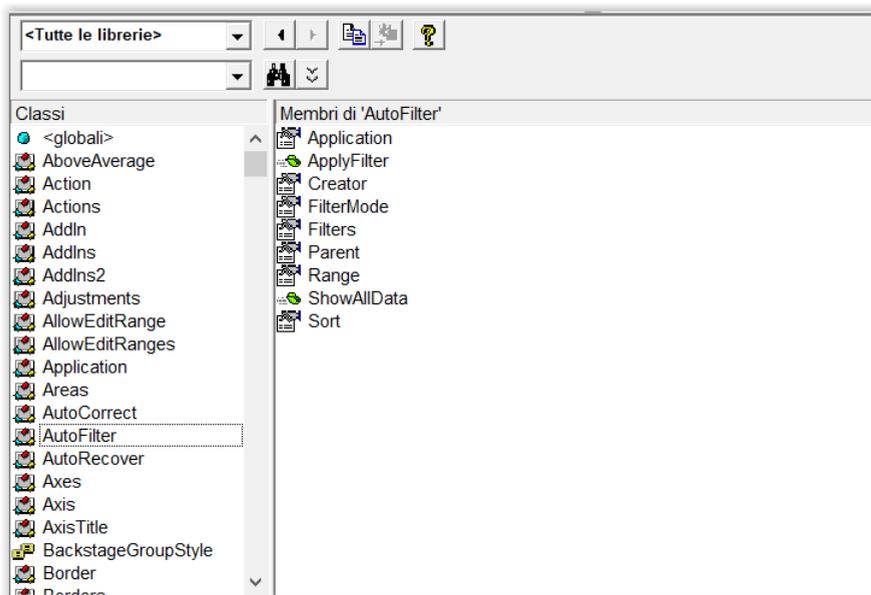


Figura 5 - Il Visualizzatore oggetti offre un modo rapido per cercare oggetti e le loro proprietà. Qui puoi vedere le proprietà contenute nell'oggetto AutoFilter.

Le informazioni sulla proprietà vengono visualizzate nella parte inferiore del Visualizzatore oggetti.

La finestra Codice

Farai la maggior parte del lavoro di testare e modificare le tue macro nella finestra Codice del Visual Basic Editor (poiché il codice è scritto in testo normale, è sufficiente scriverlo nel Blocco note e incollarlo nell'editor del codice per effettuare test e debug. Tuttavia, Visual Basic Editor offre così tanti utili strumenti di programmazione che solo pochi brillanti programmatori possono ottenere facilmente buoni risultati facendone a meno).

Il Visual Basic Editor fornisce una finestra Codice individuale per ogni progetto aperto, per ogni foglio o cartella di lavoro all'interno del progetto che possa contenere codice e per ogni modulo di codice e UserForm nel progetto. Ogni finestra Codice è identificata dal nome del progetto, dal nome del modulo all'interno del progetto e dalla parola Codice tra parentesi. La **Figura 6** mostra la finestra Codice di Visual Basic Editor con le macro `AggiungiTotale` e `AggiungiTotaleRelativo` aperte.

Come puoi vedere dalla figura, due caselle di riepilogo a discesa appaiono appena sotto la barra del titolo della finestra del codice:

- la casella di riepilogo a discesa **Oggetto** nell'angolo in alto a sinistra della finestra Codice fornisce un modo rapido per navigare tra diversi oggetti
- la casella di riepilogo a discesa **Routine** nell'angolo in alto a destra della finestra Codice consente di passare rapidamente da procedura a procedura all'interno del modulo corrente. Fai clic sul pulsante “freccia giù” per visualizzare l'elenco delle procedure. Vedrai che la prima procedura è (dichiarazioni). Facendo clic su questo elemento nell'elenco, si accede all'area Dichiarazioni nella parte superiore del foglio di codice corrente, in cui vengono dichiarate le variabili pubbliche e altre informazioni che devono essere rilevate da più procedure

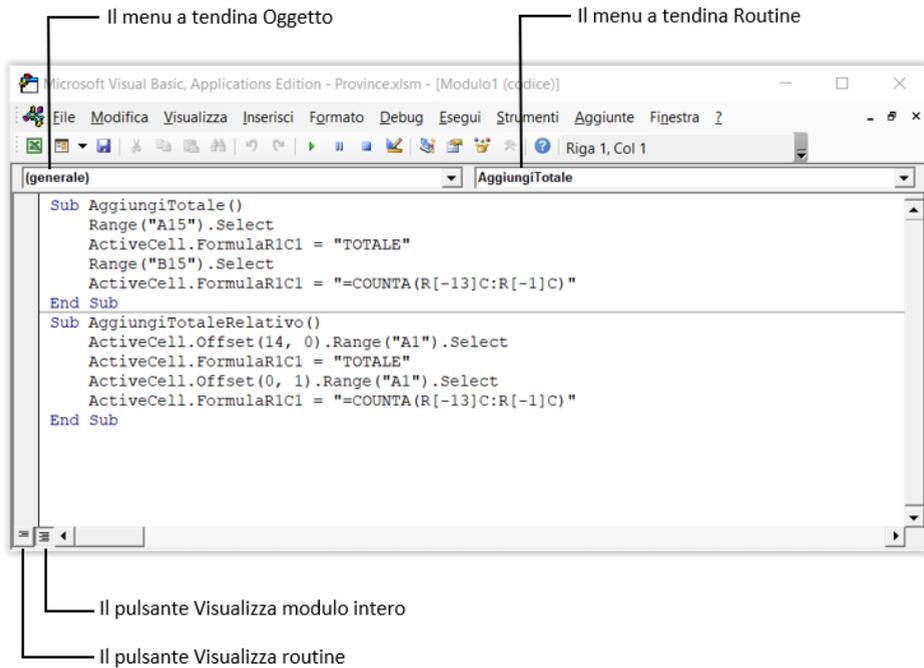


Figura 6 - Modifichi le tue macro nella finestra Codice

La finestra Codice del Visual Basic Editor offre una mezza dozzina di funzioni che consentono di modificare il codice in modo efficiente e accurato, come discusso nelle sezioni seguenti.

Auto completamento delle istruzioni

La funzione “Auto completamento” può completare la parola che stai digitando nella finestra Codice, una volta che hai digitato abbastanza lettere per distinguere quella parola da un'altra. Se non hai digitato abbastanza lettere per distinguere la parola, Visual Basic Editor ti offre l'elenco delle possibilità più

vicine (vedi **Figura 7**). Puoi continuare a digitare per restringere la selezione, o scorrere l'elenco visualizzato per trovare quello che desideri.

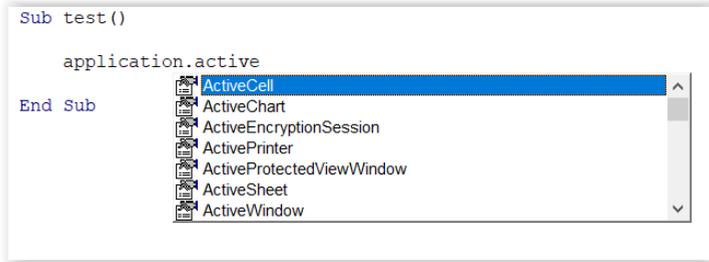


Figura 7 - La funzione Auto completamento completa automaticamente un termine quando hai digitato abbastanza per identificarlo. Se non hai digitato abbastanza, puoi scegliere da una breve lista.

Il modo più semplice per attivare l'auto completamento quando stai digitando il codice è premere **Ctrl + Barra spaziatrice**.

Puoi anche scegliere **Modifica > Completa parola** o fare clic sul pulsante **Completa parola** sulla barra degli strumenti **Modifica** (vedi **Figura 8**). Si noti che la barra degli strumenti **Modifica** non è visibile per impostazione predefinita. Puoi aprirla selezionando **Visualizza > Barre degli strumenti > Modifica** o facendo clic con il pulsante destro del mouse nell'area della barra degli strumenti dell'Editor, quindi scegliendo **Modifica** dal menu di scelta rapida visualizzato.

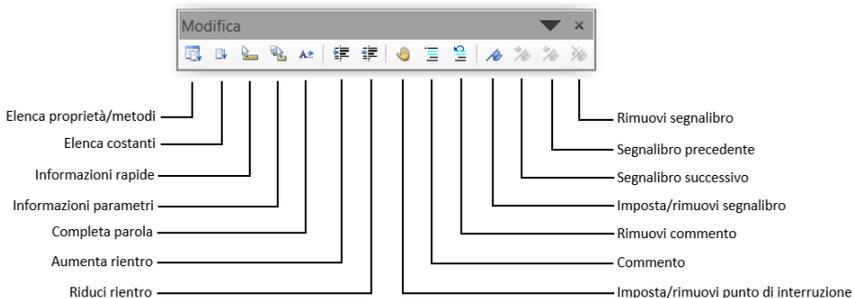
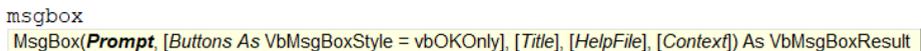


Figura 8 - La barra degli strumenti Modifica contiene funzioni utilizzate quando si lavora nella finestra Codice.

Informazioni rapide

La funzione Informazioni rapide visualizza una descrizione che mostra le informazioni sulla sintassi della variabile, funzione, metodo, comando o sub selezionato (selezionato qui significa semplicemente la parola nel codice che si trova nel punto di inserimento del cursore lampeggiante o adiacente ad esso). Se si digita un comando come **MsgBox** e quindi si preme la barra spaziatrice, viene visualizzata la descrizione per completare la digitazione nel comando. Il suggerimento mostra sia gli elementi obbligatori che quelli facoltativi di quel comando. Gli elementi opzionali sono racchiusi tra parentesi quadre.

La **Figura 9** mostra un esempio.



```
msgbox
MsgBox(Prompt, [Buttons As VbMsgBoxStyle = vbOKOnly], [Title], [HelpFile], [Context]) As VbMsgBoxResult
```

Figura 9 - Utilizza la funzione Informazioni rapide dell'Editor per visualizzare la sintassi del comando VB o una lettura rapida dello stato.

Per visualizzare le Informazioni rapide, utilizza uno di questi metodi:

- fai clic sull'icona Informazioni rapide sulla barra degli strumenti
- fai clic con il pulsante destro del mouse su un comando VB e seleziona Informazioni rapide dal menu di scelta rapida
- posiziona il punto di inserimento nel comando e premi Ctrl + I
- posiziona il punto di inserimento nel termine e seleziona Modifica ➤ Informazioni rapide
- se stai digitando comandi in VBA, il modo più semplice per visualizzare Informazioni rapide è semplicemente digitare il nome del comando (come **MsgBox**), e quindi premere il tasto della barra spaziatrice. Nota che VB non presta attenzione alla capitalizzazione delle lettere; quindi, puoi digitare msgbox o MsgBox o qualsiasi altra variazione tu voglia. Una volta che hai finito di digitare la riga di codice

(premendo Invio), l'Editor automaticamente capitalizza il comando nel modo standard: **MsgBox**

Elenco automatico membri

Molti comandi VB hanno proprietà (qualità) e metodi (comportamenti). Considerati nel loro insieme, le proprietà e i metodi di un oggetto sono chiamati i suoi membri.

Ad esempio, una *message box* può visualizzare varie icone (un punto interrogativo, un punto esclamativo e così via) per indicare all'utente lo scopo del messaggio (domanda, avvertenza, ecc.). Questa icona è chiamata proprietà Buttons dell'oggetto message-box. E questa proprietà è specificata subito dopo il messaggio di testo nella riga di codice. Pertanto, quando digito una virgola per indicare che ora sto per specificare l'icona per la mia *message box*, la funzione "elenco automatico membri" apre un elenco a discesa delle opzioni disponibili. Come puoi vedere nella **Figura 10**, sto scegliendo **vbOKOnly**, ma ci sono un certo numero di altre scelte possibili, come **vbOKCancel**, **vbQuestion** e così via.

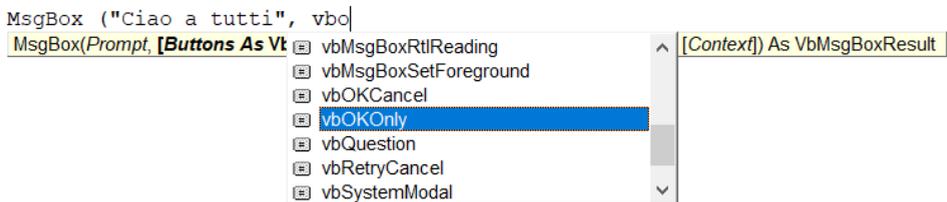


Figura 10 - Utilizzare il comando Elenco automatico membri per inserire gli elementi del codice in modo rapido e accurato.

L'elenco automatico membri consente di completare rapidamente la riga di codice. L'elenco automatico membri è attivato per impostazione predefinita e viene visualizzato automaticamente quando si digita un punto in una descrizione di oggetto o una virgola, una parentesi o altra punteggiatura in una riga di codice. Nella **Figura 10** ho digitato il comando msgbox seguito dal testo

“Ciao a tutti” e poi una virgola. Non appena ho digitato la virgola, è apparso l'elenco delle impostazioni per i pulsanti (queste impostazioni sono chiamate costanti).

In alternativa, è possibile visualizzare la casella di riepilogo facendo clic sul pulsante “Elenca proprietà/metodi” sulla barra degli strumenti Modifica.

Per utilizzare i membri dell'elenco automatico e inserirli nel codice, atteniti alla seguente procedura:

1. Premi il tasto freccia giù per scorrere in basso fino alla proprietà o al metodo che vuoi. Puoi anche digitare le prime lettere della proprietà o il nome del metodo per avvicinarti più velocemente
2. Immetti la proprietà o il metodo nel codice effettuando una delle seguenti operazioni:
 - a. Premi Tab o fai doppio clic sulla proprietà o sul metodo, se desideri continuare ad aggiungere qualcosa alla stessa riga di codice dopo aver inserito la proprietà o il metodo (potrebbero esserci proprietà opzionali aggiuntive che desideri specificare nella stessa riga)
 - b. Premi Invio se desideri iniziare una nuova riga dopo aver inserito la proprietà o il metodo

Elenco costanti

La funzione Elenco costanti visualizza una casella di riepilogo contenente le costanti di una proprietà che hai digitato, in modo da poter completare rapidamente l'espressione. L'elenco costanti è attivato per impostazione predefinita. In alternativa, è possibile visualizzare la casella di riepilogo facendo clic sul pulsante “Elenca costanti” sulla barra degli strumenti Modifica. Per utilizzare l'elenco costanti (vedi **Figura 11**), atteniti alla seguente procedura:

1. Digita “**Assistant.Animation =**” nella finestra Codice. L'elenco delle costanti verrà visualizzato come mostrato nella **Figura 11**
2. Premi ↓ (freccia giù) per scorrere verso il basso fino alla costante che stai cercando, oppure digita la prima lettera (o le prime lettere) o scorri verso il basso con il mouse
3. Immetti la costante nel codice procedendo come segue:

- Premi Tab o fai doppio clic sulla costante, se vuoi continuare a lavorare sulla stessa riga dopo aver inserito la costante
- Premi Invio se desideri iniziare una nuova riga dopo aver inserito la costante

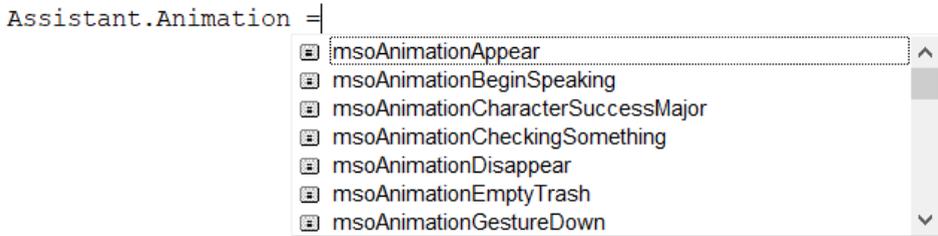


Figura 11 - La funzione Elenco costanti consente di risparmiare tempo ed energia, soprattutto quando si devono digitare nomi di costante complessi.

Descrizione dati

La funzione Descrizione dati visualizza una descrizione che contiene il valore di una variabile su cui si sposta il puntatore del mouse quando l'editor di Visual Basic si trova in modalità Interruzione (una modalità utilizzata per testare ed eseguire il debug delle macro, descritta più avanti in questo corso). La **Figura 12** mostra un esempio. La funzione Descrizione dati è attivata per impostazione predefinita, ma è possibile disattivarla, insieme alle altre funzionalità selezionando Strumenti ➤ Opzioni.

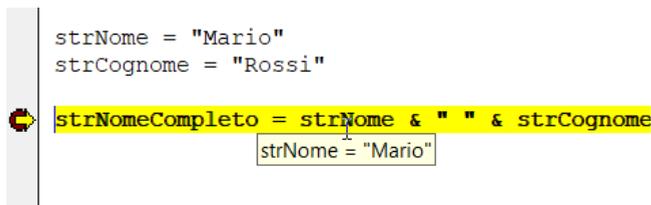


Figura 12 - Utilizza la funzione Descrizione dati per verificare il valore di una variabile quando si esegue passo-passo il codice.

Indicatori di margine

La funzione Indicatori di margine consente di impostare rapidamente un punto di interruzione, l'istruzione successiva o un segnalibro facendo clic sul margine della finestra Codice. Parleremo dell'impostazione dei punti di interruzione, dell'impostazione dell'istruzione successiva e dell'impostazione dei segnalibri in un secondo momento (puoi semplicemente fare clic con il pulsante destro sul margine grigio sul lato sinistro della finestra Codice, quindi scegliere Imposta/rimuovi dal menu di scelta rapida per modificare i punti di interruzione o i segnalibri. Puoi anche fare clic con il tasto sinistro del mouse per attivare/disattivare i punti di interruzione).

Altre funzioni di modifica

Oltre a queste funzionalità, la finestra Codice include funzioni di modifica standard di Office come copia e sposta, taglia e incolla e trascina e rilascia. È possibile trascinare il codice da una procedura o un modulo a un altro, ad esempio.

La finestra Proprietà

Il Visual Basic Editor fornisce una finestra Proprietà che è possibile utilizzare per visualizzare e modificare le proprietà di un oggetto in VBA, ad esempio un progetto, un modulo o un modulo di classe, una UserForm o un controllo (un pulsante o una casella di controllo in una finestra di dialogo, per esempio). Se la finestra Proprietà non è visibile nell'Editor, premi F4.

Nell'elenco a discesa nella parte superiore della finestra Proprietà, è possibile selezionare l'oggetto con le proprietà che vuoi visualizzare o modificare. Oppure puoi selezionare una voce nella finestra Gestione progetti. L'opzione Alfabetico visualizza un elenco alfabetico delle proprietà nell'elemento e l'opzione Per categoria presenta un elenco delle proprietà suddivise in categorie. In generale, trovo la categorizzazione poco utile perché molte proprietà non si adattano perfettamente ad una particolare categoria. La **Figura 13** mostra le proprietà di una cartella di lavoro con l'opzione Alfabetico

a sinistra e l'opzione Per categoria a destra (la visualizzazione Per categoria di una cartella di lavoro o di un foglio di lavoro di Excel non è molto utile in quanto tutte le proprietà appartengono alla categoria Varie. Non esiste alcuna classificazione qui).

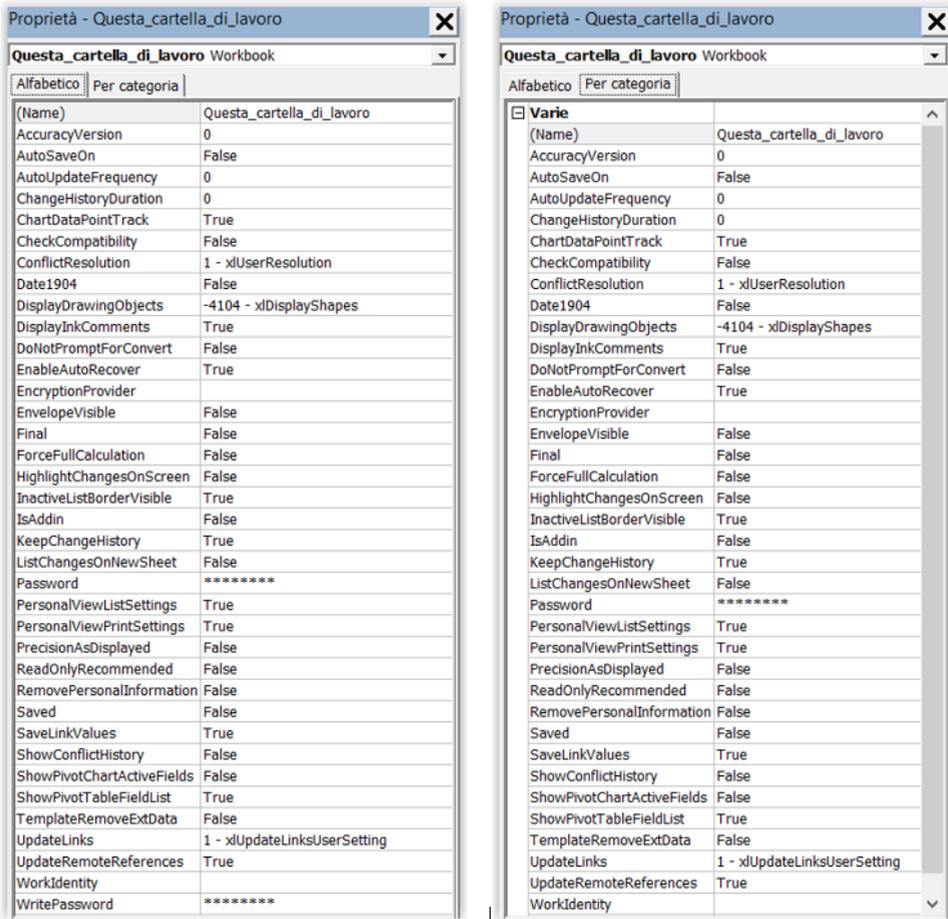


Figura 13 - Utilizza la finestra Proprietà per visualizzare le proprietà di un progetto, UserForm, modulo, modulo di classe o controllo.

Lo scopo della maggior parte delle proprietà della cartella di lavoro è facile da comprendere. Ad esempio, se la proprietà **Saved** è impostata su **False**, ciò indica che la cartella di lavoro contiene modifiche non salvate.

Differenze tra modalità di progettazione, modalità di esecuzione e modalità di interruzione

L'editor di Visual Basic può essere in una delle tre modalità, rispecchiando tre fasi fondamentali della programmazione: scrittura del codice, individuazione di un bug e risoluzione di un bug:

Modalità di progettazione, conosciuta anche come *design time*. Ogni volta che stai lavorando nel Visual Basic Editor sul tuo codice, sei in modalità Progettazione. Non è necessario che tu stia progettando attivamente qualcosa visivamente, come un controllo utente o un modulo, anche se lo farai spesso. Spesso digiterai anche il codice sorgente, ovvero i comandi che Visual Basic eseguirà quando passerai in modalità Esecuzione, oppure potresti modificare il codice che hai registrato.

Modalità di esecuzione, conosciuta anche come *runtime*. Quando il codice è in esecuzione, sei in modalità Esecuzione. La macro verrà eseguita come se fosse stata lanciata da Excel (utilizzando una combinazione di tasti di scelta rapida o facendo clic sul pulsante della barra di accesso rapido). Lo scopo della modalità Esecuzione nel Visual Basic Editor è consentire all'utente di testare e osservare il comportamento del codice e interagire con esso, se necessario, per verificare che funzioni come previsto. Questa fase è conosciuta come *debugging*. Se si riscontrano problemi durante i test di *runtime*, è possibile interrompere l'esecuzione premendo Ctrl + Interr e quindi controllare i valori nelle variabili o tentare in altro modo di rintracciare dove si trova l'errore nel codice. VBA stesso può anche lanciarti in modalità Interruzione se rileva una condizione di errore.

Modalità interruzione. Quando il codice è in esecuzione ma l'esecuzione è temporaneamente sospesa, sei in modalità Interruzione. Tra le altre cose, la modalità Interruzione ti consente di eseguire il codice un comando o una

procedura alla volta (anziché eseguire tutti i comandi contemporaneamente alla massima velocità). L'esecuzione passo-passo è uno strumento molto utile quando esegui il debug o quando stai comunque analizzando il tuo codice. Esplorerai in dettaglio le tecniche di debug nella sezione relativa alla Gestione degli errori.

Per modificare una proprietà, fai clic sulla cella contenente il nome della proprietà. Se nella cella del valore viene visualizzato un pulsante con una freccia in giù, fai clic per scegliere un nuovo valore da un elenco a discesa. Se non viene visualizzato alcun pulsante, fai clic sulla cella del valore per visualizzare il cursore di inserimento lampeggiante per poi digitare un nuovo valore.

Potrai scegliere valori diversi dagli elenchi a discesa, a seconda del tipo di proprietà. Per una proprietà True/False, sarai limitato a queste due scelte nell'elenco a discesa. Per una proprietà testuale come Name, è possibile inserire qualsiasi nome VBA valido.

Per impostazione predefinita, la finestra Proprietà è ancorata sotto Gestione progetti. È possibile regolare le altezze relative della finestra Proprietà o della finestra Gestione progetti trascinando il bordo tra di esse. Puoi anche allargare entrambe contemporaneamente trascinando il bordo alla loro destra. Se si disancora la finestra Proprietà (trascinandola o facendo doppio clic sulla barra del titolo), è possibile ridimensionarla trascinandone i bordi o gli angoli per visualizzare più proprietà o per restringere la finestra in modo che occupi meno spazio nel Visual Basic Editor. Tutte le finestre interne (chiamate anche riquadri, ad esempio il riquadro Proprietà) possono essere disancorate trascinandole per la barra del titolo o facendo doppio clic sulla barra del titolo. Sempre facendo doppio clic sulla barra del titolo, è poi possibile ancorarle di nuovo.

La finestra Immediata

Oltre a Gestione progetti e alla finestra Codice, il Visual Basic Editor include alcune altre finestre che non viene visualizzato per impostazione predefinita. Tre di queste finestre sono molto importanti: la finestra Proprietà e il Visualizzatore oggetti (entrambi descritti in precedenza in questo capitolo) e la finestra Immediata, mostrata nella **Figura 14**. La finestra Immediata è una piccola finestra disadorna che puoi usare come un gratta e vinci virtuale per inserire linee di codice che vuoi testare senza inserirle in una macro reale. Quando si digita una riga di codice nella finestra Immediata e si preme il tasto Invio, il Visual Basic Editor esegue quel codice. Utilizzerai la finestra Immediata durante la discussione del linguaggio VBA nella sezione "Nozioni fondamentali sulla sintassi VBA".

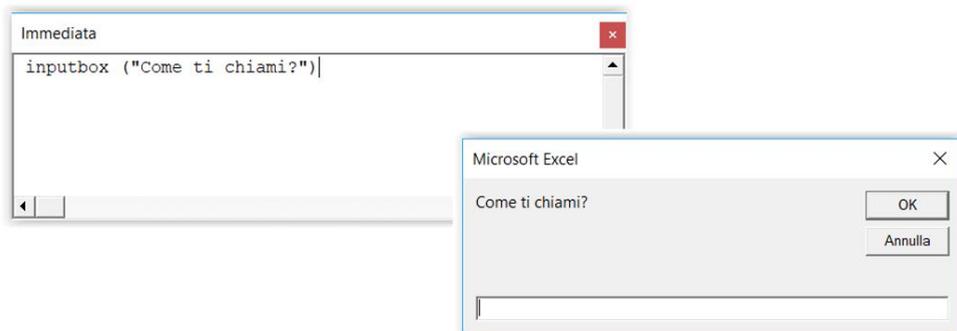


Figura 14 - Utilizza la finestra Immediata per lavorare e ottenere informazioni in tempo reale.

Per visualizzare la finestra Immediata, premi **Ctrl + G** o seleziona **Visualizza** ➤ **Finestra Immediata**.

Visualizza lo stato delle variabili durante il debug

È inoltre possibile utilizzare la finestra Immediata per visualizzare informazioni che consentono di verificare i valori di variabili ed espressioni mentre il codice è in esecuzione. Questo viene fatto usando il comando **Debug.Print**, come in questo esempio, che mostra il valore della variabile x nella finestra Immediata:

```
Sub MostraDebug()  
  
Dim x As Integer  
x = 12  
  
Debug.Print x  
  
End Sub
```

Impostazione delle proprietà di un progetto

Ogni progetto VBA ha diverse proprietà che è possibile impostare, incluso il nome del progetto, la sua descrizione e se è protetto in visualizzazione. Per esaminare o impostare le proprietà di un progetto, fai clic con il tasto destro del mouse sul progetto o su uno dei suoi componenti in Gestione progetti e scegli la voce “Proprietà progetto” nel menu di scelta rapida. Si aprirà la finestra di dialogo Proprietà progetto.

Sia la voce di menu che la finestra di dialogo risultante sono identificati dalla descrizione del progetto, ad esempio, la finestra di dialogo delle proprietà per una cartella di lavoro di Excel è identificata come VBAProject - Proprietà progetto. La **Figura 15** ne mostra un esempio.

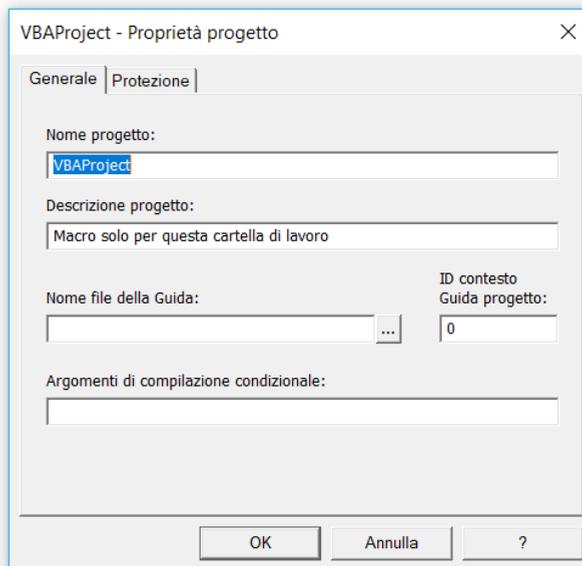


Figura 15 - Utilizza la finestra di dialogo Proprietà progetto per visualizzare e impostare le proprietà di un progetto e proteggere un progetto dalle modifiche.

Ecco cosa puoi fare nella scheda Generale della finestra di dialogo Proprietà progetto:

- Impostare il nome del progetto nella casella di testo Nome progetto. Questo nome identifica il progetto nel Visualizzatore oggetti e, se necessario, nel registro di Windows. Assicurati che il nome sia univoco per evitare confusione con qualsiasi altro progetto. Tecnicamente, il nome del progetto è il nome della libreria dei tipi per il progetto (una libreria dei tipi descrive gli oggetti, come moduli e UserForms, che il progetto contiene); è usato per costruire il nome di classe completo delle classi nel progetto (ne parleremo più avanti nel corso). Il nome del progetto può contenere caratteri di sottolineatura, ma non può contenere spazi
- Immettere una descrizione del progetto nella casella di testo Descrizione progetto. Questa descrizione viene visualizzata nel riquadro Descrizione nel Visualizzatore oggetti per aiutare l'utente a

capire qual è il progetto. Quindi, deve essere concisa, ma la più descrittiva possibile

- Designare il file di aiuto per il progetto immettendo il nome e il percorso del file della Guida nella casella di testo Nome file della guida. Fai clic sul pulsante contrassegnato con i puntini di sospensione (...) a destra della casella di testo Nome file della guida per visualizzare la finestra di dialogo File della guida. Quindi seleziona il file e fai clic sul pulsante Apri per immettere il nome del file della Guida nella casella di testo (in alternativa, puoi digitare o incollare il nome e il percorso)
- Specificare il contesto della Guida per il progetto nella casella di testo ID contesto Guida progetto. Il contesto della Guida fa riferimento a una posizione nel file della Guida. Il contesto predefinito della Guida è 0, che fa sì che il file della Guida visualizzi la sua schermata di apertura (la stessa schermata verrà visualizzata se si esegue il file della Guida dalla finestra di dialogo Esegui o facendo doppio clic sul file in Esplora risorse). È possibile specificare un contesto di aiuto diverso per portare l'utente a un particolare argomento, ad esempio un altro più pertinente al progetto del quale stanno cercando aiuto
- Specificare qualsiasi argomento di compilazione condizionale necessario per il progetto. Alcuni trovano utili le compilazioni condizionali, ma la maggior parte no. Ti consente di creare diverse versioni di un progetto VBA in base a ciò che specifichi in questa finestra di dialogo. Personalmente, trovo che generi più problemi che benefici, e crei anche confusione. Invece preferirei creare più di una versione di un progetto. Se sei interessato a questo argomento, dai un'occhiata qui:

<https://docs.microsoft.com/it-it/dotnet/visual-basic/programming-guide/program-structure/conditional-compilation>

Ecco cosa puoi fare nella scheda Protezione della finestra di dialogo Proprietà progetto, mostrata in **Figura 16**:

Selezionare la casella di controllo “Proteggi progetto dalla visualizzazione” per impedire ad altre persone di aprire il progetto, visualizzarlo e modificarlo senza conoscere la password

Nel gruppo “Password per visualizzare le proprietà del progetto” nella casella **Password**, immetti una password; quindi, inserisci la stessa password nella casella di testo **Conferma password**. Fai clic sul pulsante OK e quindi chiudi il progetto. Ora nessuno può aprire e visualizzare (e tanto meno modificare) il progetto se non conosce la password. Detto questo, la sicurezza delle password di Office è stata debole ed era facile violarla prima di Office 2007. Ora vengono utilizzate tecniche di crittografia superiori, ma la password è ancora crackabile, anche se con una maggiore difficoltà. Maggiori informazioni al riguardo nella dispensa 39 "Le funzionalità di sicurezza di VBA".

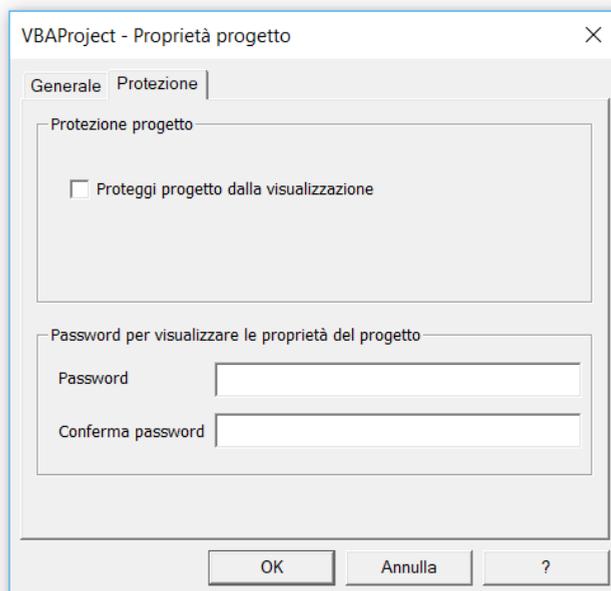


Figura 16 - La scheda Protezione della finestra di dialogo Proprietà progetto consente di bloccare il progetto con una password in modo che nessuno possa visualizzarlo o modificarlo

Seleziona Proteggi progetto dalla visualizzazione se vuoi impedire ad altri di aprirlo

Se si inserisce una password nella casella di testo Password e nella casella di testo Conferma password ma non si seleziona la casella di controllo “Proteggi progetto dalla visualizzazione”, il Visual Basic Editor richiederà la password la prossima volta che tenterai di visualizzare la finestra di dialogo Proprietà progetto. Tuttavia, sarai in grado di aprire e visualizzare il progetto e i suoi contenuti senza fornire la password.

Excel VBA Masterclass

by Excel Academy (<https://www.excelacademy.it/>)

Questa dispensa è parte del corso online “Excel VBA Masterclass”

(<https://www.excelacademy.it/courses/excel-vba-masterclass/>)

Copyright © 2023 Excel Academy. Tutti i diritti sono riservati.

Realizzato in Italia.

Nessuna parte di questa dispensa deve essere riprodotta, salvata online per una libera consultazione o trasmessa con qualsiasi mezzo, elettronico, meccanico, tramite fotocopia, registrazione o altro, senza autorizzazione scritta dell'autore.

Sebbene sia stata presa ogni precauzione nella preparazione di questa dispensa, l'autore non si assume alcuna responsabilità per errori o omissioni. Inoltre, non si assume alcuna responsabilità per danni derivanti dall'utilizzo delle informazioni qui contenute.