*usr_42.txt* Per **Vim version 8.1.** Ultima modifica: 2008 May 05

VIM USER MANUAL - di Bram Moolenaar
Traduzione di questo capitolo: Luca Ferraro

# Aggiungere nuovi menù

A questo punto sapete che Vim è molto flessibile. Ciò include i menù usati nella GUI. Potete definire le vostre proprie opzioni di menù per rendere alcuni comandi più facilmente accessibili. Cito è possibile solo per utenti felici di usare il mouse!



==========

## *42.1* Introduzione

I menù che Vim usa sono definiti nel file "$VIMRUNTIME/menu.vim". Se volete scrivere i vostri menù personali, potreste prima dare un'occhiata a questo file.

Per definire un elemento del menù, utilizzate il comando ":menu". La forma fondamentale di questo comando è molto semplice: >

```
:menu {elemento-menù} {comando}
```

L' {elemento-menù} descrive in quale posizione del menù inserire l'elemento. Un tipico {elemento-menù} è "File.Save", che rappresenta l'elemento "Save" sotto il menù "File". Un punto viene utilizzato per separare i nomi. Un esempio: >

```
:menu File.Save  :update<CR>
```

Il comando ":update" salva il file quando esso sia stato modificato.

Potete aggiungere un altro livello: "Edit.Settings.Shiftwidth" definisce un sotto menù "Settings" nel menù "Edit", con l'elemento "Shiftwidth". Potreste utilizzare anche livelli inferiori. Non usatelo troppo, vi toccherebbe spostare il mouse un bel po' per poter usare così un elemento.

Il commando ":menu" è molto simile al comando ":map" : il lato sinistro specifica quale elemento verrà lanciato ed il destro definisce i caratteri che verranno eseguiti. {keys} sono caratteri, vengono usati solo come li avevate scritti. Così nel modo Insert, quando {keys} è puro testo, questo testo viene inserito.

### TASTI DI SCELTA RAPIDA

Il carattere ampersand (&), viene utilizzato per indicare un tasto di scelta rapida. Per esempio, potete utilizzare Alt-F per selezionare il menù "File" e quindi premere S per il comando "Save". (L'opzione 'winaltkeys' può peraltro disabilitare questa possibilità!). Comunque il {menù-item} appare come "&File.&Save". I caratteri acceleratori verranno sottolineati nel menù.

Dovrete fare attenzione che ogni tasto venga usato una sola volta in ciascun menù. Altrimenti non sapreste quale dei due potrebbe venire usato. Vim non vi avvisa di ciò.

### PRIORITA'

La definizione corrente dell'elemento di menù File.Save è come segue: >

```
:menu 10.340 &File.&Save<Tab>:w  :confirm w<CR>
```

Il numero 10.340 è chiamato il numero di priorità. Questo viene utilizzato dall'editor per decidere in quale posizione inserire l'elemento di menù. Il primo numero (10) indica la posizione sulla barra dei menù. I menù con i

valori più bassi vengono posizionati a sinistra, quelli con valori più alti a destra.
   Questi sono le priorità usate per i menù standard:

```
        10    20     40     50      60       70               9999

      +-------------------------------------------------------------+
      | File  Edit  Tools  Syntax  Buffers  Window            Help |
      +-------------------------------------------------------------+
```

Notate che al menù Help viene dato un numero molto grande, per farlo apparire il più a destra possibile.
   Il secondo numero (340) determina la posizione dell'elemento nel menù a discesa.  I numeri più bassi vanno in alto, quelli più alti in fondo.  Queste sono le priorità nel menù File:

```
                  +-----------------+
        10.310    |Open...          |
        10.320    |Split-Open...    |
        10.325    |New              |
        10.330    |Close            |
        10.335    |---------------- |
        10.340    |Save             |
        10.350    |Save As...       |
        10.400    |---------------- |
        10.410    |Split Diff with  |
        10.420    |Split Patched By |
        10.500    |---------------- |
        10.510    |Print            |
        10.600    |---------------- |
        10.610    |Save-Exit        |
        10.620    |Exit             |
                  +-----------------+
```

Notate che c'è un intervallo tra i numeri.  Qui è dove potete inserire i vostri elementi personali, se lo volete davvero (spesso è meglio lasciare stare i menù standard ed aggiungere un nuovo menù per i vostri elementi personali).
   Quando create un sotto-menù, potete aggiungere un altro ".numero" alla priorità.  Così ogni nome nel {menù-item} ha i propri numeri di priorità.

## CARATTERI SPECIALI

Il {menù-item} in questo esempio è  "&File.&Save<Tab>:w".  Ciò evidenzia un punto importante: {menù-item} deve essere una sola parola.  Se volete mettere un punto, spazio o tabulatore nel nome usate la notazione <> (<space> e <tab>, ad esempio) o premettere al carattere un backslash (\). >

```
        :menu 10.305 &File.&Do\ It\.\.\. :exit<CR>
```

In questo esempio, l'elemento di menù "Do It..." contiene uno spazio ed il comando è ":exit<CR>".

Il carattere <Tab> nel nome del menù viene usato per separare la parte che definisce il nome del menù da quella che da un'indicazione all'utente.  La parte dopo il <Tab> viene visualizzata allineata nel menù.  Nel menù File.Save il nome usato è "&File.&Save<Tab>:w".  Così il nome del menù è File.Save e l'indicazione è ":w".

## SEPARATORI

Le linee di separazione, utilizzare per raggruppare assieme elementi di menù, possono essere definite usando un nome che inizi e finisca in un '-'.  Ad esempio "-sep-".  Se si usano diversi separatori i nomi debbono essere differenti.  Altrimenti i nomi non vanno.
   Il comando da un separatore non verrà mai eseguito, ma voi dovete definirne comunque uno.  Un solo due punti va bene.  Esempio: >

```
        :amenu 20.510 Edit.-sep3- :
```

==========

*42.2*  Comandi dei menù

Potete definire elementi del menù che esistono solo per alcuni modi Vim. Ciò funziona proprio come le variazioni nel comando ":map" :

| | |
|---|---|
| :menu | Modo Normal, Visual e Operator-pending |
| :nmenu | modo Normal |
| :vmenu | modo Visual |
| :omenu | modo Operator-pending |
| :menu! | modo Insert e Command-line |
| :imenu | modo Insert |
| :cmenu | modo Command-line |
| :tlmenu | modo Terminal |
| :amenu | Tutti i modi (tranne il modo Terminal) |

Per evitare che i comandi di un elemento del menù vengano mappati, usate il comando ":noremenu", ":nnoremenu", ":anoremenu", etc.

## UTILIZZO DI :AMENU

Il comando ":amenu" è leggermente diverso. Esso assume che le {keys} che voi date debbano essere eseguite in modo Normal. Quando Vim è in modo Visual o in modo Insert quando il menù viene usato, Vim prima deve tornare al modo Normal.
":amenu" inserisce un CTRL-C o un CTRL-O per voi. Per esempio, se usate questo comando: >

```
:amenu  90.100 Mine.Find\ Word  *
```

Allora i comandi del menù risultanti saranno:

| | |
|---|---|
| modo Normal: | * |
| modo Visual: | CTRL-C * |
| modo Operator-pending: | CTRL-C * |
| modo Insert: | CTRL-O * |
| modo Command-line: | CTRL-C * |

Se siete in modo Command-line, il CTRL-C abbandonerà il comando sinora scritto. Nei modi Visual e Operator-pending CTRL-C interromperà il modo. In modo Insert CTRL-O eseguirà il comando e quindi ritornerà in modo Insert.
CTRL-O funziona per un solo comando. Se avete necessità di usare due o più comandi, inseriteli in una funzione e quindi chiamate questa funzione. Esempio: >

```
:amenu  Mine.Next\ File  :call <SID>NextFile()<CR>
:function <SID>NextFile()
:  next
:  1/^Code
:endfunction
```

Questa opzione del menù va al prossimo file nella lista degli argomenti con ":next". Poi cerca la linea che inizia con "Code".
Il <SID> prima del nome della funzione è l'ID dello script. Ciò rende la funzione locale allo script di Vim corrente. Ciò evita problemi quando una funzione con lo stesso nome viene definita in un altro file di script. Vedere |**<SID>**|.

## MENÙ SILENTI

Il menù esegue le {keys} come le avete scritte. Per un comando ":" ciò significa che vedrete il comando scritto sulla linea di comando. Se è un comando lungo, allora apparirà il prompt hit-Enter. Ciò potrebbe essere molto fastidioso!
Per evitare ciò rendete muto il menù. Ciò viene fatto con l'argomento <silent>. Ad esempio, prendete al chiamata a NextFile() nell'esempio precedente. Quando usate questo menù vedrete ciò sulla linea di comando:

```
:call <SNR>34_NextFile() ~
```

Per evitare la comparsa del testo, inseriamo "<silent>" come primo argomento: >

```
:amenu <silent> Mine.Next\ File :call <SID>NextFile()<CR>
```

Non utilizzate "<silent>" troppo spesso. Non è necessario per comandi brevi. Se realizzate dei menù per qualcun altro, poter vedere il comando eseguito darà un suggerimento su ciò che egli può avere scritto, invece di usare il mouse.

ELENCARE I MENU'

Quando un comando del menù viene usato senza la parte {keys}, esso elenca i menù già definiti. Potete specificare un questo si comporta richiamando il contenuto di un menù già definito. Potete specificare un {menù-item} od una parte di esso, per elencare menù specifici. Esempio: >

```
:amenu
```

Ciò elenca tutti i menù. Che è una lista lunga! Meglio specificate il nome di un menù per ottenere una lista più corta: >

```
:amenu Edit
```

Questo mostra solo gli elementi del menù "Edit" per tutte i modi Vim. Per mostrare solo uno specifico elemento del menù per il modo Insert: >

```
:imenu Edit.Undo
```

Attenti a scrivere esattamente il nome giusto. Qui contano maiuscole e minuscole. Ma l'&' per gli acceleratori può venire omesso. Il <Tab> e ciò che viene dopo può ugualmente venir lasciato fuori.

CANCELLARE I MENU'

Per cancellare un menù, si utilizza lo stesso comando usato per elencarli, ma con "menu" cambiato in "unmenu". Così ":menu" diventa ":unmenu", ":nmenu" diventa ":nunmenu", ecc. Per eliminare l'elemento "Tools.Make" per il modo Insert: >

```
:iunmenu Tools.Make
```

Potete cancellare un intero menù, con tutti i propri elementi, usando il nome del menù stesso: Esempio: >

```
:aunmenu Syntax
```

Ciò cancella il menù Syntax e tutti i suoi elementi.

==========

*42.3* Varie

Potete cambiare l'aspetto dei menù tramite dei flag in 'guioptions'. Nel valore di default esse sono tutte incluse, tranne "M". Potete rimuovere un flag con un comando del tipo: >

```
:set guioptions-=m
```

<

| | |
|---|---|
| m | Quando rimossa la barra dei menù non è visualizzata. |
| M | Quando aggiunta non sono caricati i menù di default. |
| g | Quando rimossa i menù inattivi sono completamente rimossi (Non funziona su tutti i sistemi). |
| t | Quando rimossa il "tearoff" non viene abilitato. |

La linea punteggiata in cima ai menù non è una linea di separazione. Quando selezionate questo elemento il menù viene "teared-off": Viene mostrato entro una finestra separata. Questo viene chiamato menù tearoff. È utile quando usate lo stesso menù spesso.

Per la traduzione degli elementi dei menù, vedete |**:menutrans**|.

Poiché bisogna utilizzare il mouse per selezionare gli elementi dei menù, è

bene usare il comando ":browse" per selezionare un file. Ed il comando ":confirm" per avere una finestra di dialogo invece di un messaggio di errore, ad esempio, quando il buffer corrente ha subito modifiche. Questi due possono venir combinati: >

```
:amenu File.Open  :browse confirm edit<CR>
```

":browse" fa apparire un navigatore di file per selezionare il file da aprire. ":confirm" fa apparire una finestra di dialogo se il file corrente ha subito modifiche. Potete scegliere se salvare le modifiche, gettarle via o cancellare il comando.

Per elementi più complicati, le funzioni confirm() e inputdialog() possono venir utilizzate. I menù di default contengono qualche esempio.

==========

*42.4* Toolbar ed i menù popup

Ci sono due menù speciali: ToolBar e PopUp. Elementi che iniziano con questi nomi non compaiono nella barra del menù normale.

TOOLBAR

La toolbar compare solo se il flag "T" è inclusa nell'opzione 'guioptions'.

La toolbar utilizza icone piuttosto che testo per rappresentare il comando. Per esempio, il {menù-item} chiamato "ToolBar.New" provoca la comparsa dell'icona "New" sulla toolbar.

Vim ha un set di 28 icone built-in. Potete trovare una tabella qui: |**builtin-tools**|. La maggior parte di queste è già usata nella toolbar di default. Potete ridefinirne l'azione (dopo che i menù di default siano a punto).

Potete aggiungere una nuova bitmap per un elemento della toolbar. Oppure definire un nuovo elemento per la toolbar con una bitmap. Per esempio, definiamo un nuovo elemento di toolbar con: >

```
:tmenu ToolBar.Compile  Compila il file corrente
:amenu ToolBar.Compile  :!cc %:S -o %:r:S<CR>
```

Ora bisogna creare una nuova icona. Per gli utenti di MS-Windows, questa deve essere in formato bitmap, con il nome "Compile.bmp". Per Unix viene usato il formato XPM, il nome è "Compile.xpm". Le dimensioni devono essere 18x18 pixel (sotto Windows è possibile utilizzare anche altre dimensioni, ma l'aspetto è poco elegante).

Mettete le bitmap nella directory "bitmaps" entro una delle directory del 'runtimepath'. Ad esempio, per Unix "~/.vim/bitmaps/Compile.xpm".

Potete definire anche delle didascalie per le icone della toolbar. Una didascalia è una breve testo che spiega cosa farà un elemento della toolbar. Per esempio "Open file". Apparirà quando il puntatore del mouse si trova sull'elemento, senza spostarsi per un attimo. Ciò è molto utile per capire l'immagine se non fosse chiara. Esempio: >

```
:tmenu ToolBar.Make  Run make in the current directory
<
```

Nota:
Fate molta attenzione alle maiuscole/minuscole. "Toolbar" e "toolbar" sono diverse da "ToolBar"!

Per eliminare una didascalia, usate il comando |**:tunmenu**|.

L'opzione 'toolbar' può essere utilizzata per mostrare del testo invece di una bitmap, oppure entrambi testo ed una bitmap. Molti usano solo la bitmap poiché il testo prende più spazio.

MENU' POPUP

I menù popup compaiono dove si trova il mouse. In MS-Windows li attivate facendo clic con il tasto destro del mouse. Allora potete selezionare un elemento con il tasto sinistro del mouse. In Unix i menù popup si attivano facendo clic col tasto destro del mouse e poi rilasciandolo.

I menù popup compaiono solo se l'opzione 'mousemodel' è stata impostata

su "popup" o su "popup_setpos". La differenza tra le due è che "popup_setpos" muove il cursore verso la posizione in cui si trova il puntatore del mouse. Quando si fa clic entro una selezione, questa verrà utilizzata non modificata. Quando c'è una selezione, ma fate clic fuori di essa, la selezione viene rimossa.
Esistono menù popup separati per ogni modo Vim. Così non ci sono mai elementi inattivi come nei menù normali.

Qual è il significato della vita, dell'universo e di ogni cosa? *42*
Douglas Adams, la sola persona che ha conosciuto cosa fosse davvero questa questione è ormai morto, sfortunatamente. Così adesso potreste chiedervi quale sia il significato della morte...

==========

Capitolo seguente: |**usr_43.txt**| Utilizzo dei tipi di file



Per segnalazioni scrivere a vimdoc.it at gmail dot com
oppure ad Antonio Colombo azc100 at gmail dot com