

helphelp.txt Per Vim version 8.1. Ultima modifica: 2019 Oct 18

VIM Manuale di Riferimento di Bram Moolenaar
Traduzione di questo testo: Antonio Colombo

Aiuto per i file di help

helphelp

1. Comandi di help
2. File di help tradotti
3. Scrittura file di help

|online-help|
|help-translated|
|help-writing|

1. Comandi di help

online-help

<Help>
:h[elp]

o

help *<Help>* *:h* *:help* *<F1>* *i_<F1>* *i_<Help>*

Apre una finestra e visualizza il file di help nel modo di Sola lettura. Se c'è già una finestra di help aperta, questa viene riutilizzata. Altrimenti, se la finestra corrente usa la larghezza intera dello schermo, oppure è larga almeno 80 caratteri, la finestra di help sarà aperta sopra quella corrente. Altrimenti la nuova finestra è aperta nella posizione più alta della schermata. L'opzione 'helplang' è usata per scegliere la lingua, se il file principale di aiuto è disponibile in più lingue.

:h[elp] {argomento}

{subject} *E149* *E661*

Come ":help", ma in più ci si posiziona al tag {argomento}.
Per esempio: >
:help options

<

{argomento} può includere metacaratteri come "*", "?" e "[a-z]":
:help z? va a help per qualsiasi comando "z"
:help z. va a help per "z."
Ma nel caso un tag con esattamente quel nome esista, è a questo che si salta:
:help :? va a help per ":@"

Se non c'è una corrispondenza esatta per l'espressione, o se c'è più di una corrispondenza, quella "migliore" verrà usata. Un sofisticato algoritmo è usato per decidere quale sia la migliore corrispondenza. Questi fattori sono considerati nella valutazione:

- Una corrispondenza con la stessa grafia (maiuscola o minuscola) è molto meglio che una corrispondenza con differenza fra maiuscole e minuscole.
- Una corrispondenza che inizia dopo un carattere non alfanumerico è meglio di una che si trovi in mezzo a una parola.
- Una corrispondenza all'inizio o presso l'inizio di un tag è meglio di una corrispondenza più avanti.
- Quanto maggiore è il numero di caratteri alfanumerici che corrispondono, tanto migliore è la corrispondenza.
- Quanto più corta è la corrispondenza, tanto meglio.

L'opzione 'helplang' è usata per scegliere una lingua, se {argomento} è disponibile in più lingue. Per richiedere un tag in un particolare linguaggio, appendere "@ab", laddove "ab" è il codice di due lettere che indica il linguaggio. Vedere |help-translated|.

Nota Quanto più lungo l'{argomento} immesso, minore sarà il numero di corrispondenze trovate. Ci si può fare un'idea di come funziona la cosa usando il

```

completamento della linea di comando (battere CTRL-D
dopo ":help argomento" |c_CTRL-D|).
Se ci sono più corrispondenze, se ne può avere una
lista immettendo CTRL-D. Per esempio: >
    :help cont<Ctrl-D>
< Invece che battere ":help CTRL-V" per chiedere l'help
relativo a CTRL-V, si può battere: >
    :help ^V
<
Questo si può fare anche in presenza di altri
caratteri, per esempio, per chiedere l'help relativo
a CTRL-V in Modo Insert, potete battere: >
    :help i^V
<
È anche possibile iniziare a battere ":help", e poi
usare ":tag {pattern}" nella finestra di help.
Il comando ":tnext" si può quindi usare per andare
a vedere altre corrispondenze, e "tselect" per
elencare corrispondenze fra cui scegliere. >
    :help index
    :tselect /.mode
<
Quando non si immette un argomento, vengono
visualizzate le corrispondenze per "help", per evitare
una lista di TUTTE le possibili corrispondenze (questo
rallenterebbe parecchio la sessione di edit).
Il numero di corrispondenze visualizzate è limitato a
un massimo di 300.

Il comando `:help` può essere seguito da '|' e da un
altro comando, ma non serve "proteggere" (con un "\")
il '|' all'interno del comando di help. Quindi
entrambi questi comandi funzionano: >
    :help |
    :help k| only
<
Nota Lo spazio prima di '|' è interpretato come parte
dell'argomento per ":help".
Si possono anche usare <LF> o <CR> per separare il
comando di help da un comando successivo. È
necessario immettere CTRL-V prima di poter inserire
il <LF> o il <CR>. Esempio: >
    :help so<C-V><CR>only

:h[elp]! [argomento] Come ":help", ma nel caso si stia usando un file di
help diverso da quello in inglese, viene preferita una
tag trovata in un file che sia scritto nello stesso
linguaggio del file correntemente in uso. Vedere
|help-translated|.

*:helpc* *:helpclose*
:helpc[lose] Chiude una finestra di help, se ce n'è una.
Vim tenta di ripristinare la disposizione precedente
della finestra (incluso il posizionamento del
cursore) allo stato in cui era prima dell'apertura
iniziale della finestra di help. Ciò potrebbe
implicare l'esecuzione di parecchi autocomandi.

*:helpg* *:helpgrep*
:helpg[rep] {espressione}[@xx]
Viene fatta una ricerca in tutti i file di testo di
help e generata una lista in cui sono contenute le
corrispondenze a {espressione}. Si va poi a
visualizzare la prima corrispondenza.
Il parametro opzionale [@xx] richiede di trovare
solo le corrispondenze relative al linguaggio "xx".
È possibile visitare tutte le corrispondenze usando
i comandi |quickfix|, per esempio, |:cnex| per
passare alla prossima corrispondenza. Oppure si può
usare |:cwindow| per visualizzare la lista delle
corrispondenze nella finestra di quickfix.
{espressione} è valutata come un'espressione regolare
(regex) di Vim |pattern|.
Il valore di 'ignorecase' non è usato, aggiungere

```

```

"\" per ignorare maiuscolo/minuscolo nei confronti.
Esempio di ricerca che considera maiuscolo/minuscolo: >
:helpgrep Uganda
< Esempio di ricerca che ignora maiuscolo/minuscolo: >
:helpgrep uganda\"
< Esempio di ricerca nell'help in francese: >
:helpgrep backspace@fr
< L'espressione di ricerca non prevede l'uso di
interruzioni di linea, deve quindi essere soddisfatta
all'interno di una singola linea. È possibile, in
alternativa, usare |:grep|, ma è necessario per
questo procurarsi la lista dei file di help, il che
non è semplice.
Questo comando non può essere seguito da un altro
comando, qualsiasi cosa venga specificata è
considerata come facente parte dell'espressione
regolare ricercata. Ma è possibile usare |:execute|
se necessario.
I file di help che sono compressi non saranno usati
nella ricerca (Fedora comprime i file di help).

```

```
*:lh* *:lhelpgrep*
```

```
:lh[elpgrep] {pattern}[@xx]
Come ":helpgrep", tranne che la lista locazioni
viene usata al posto della lista di quickfix. Se la
finestra di help è già aperta, allora viene usata la
lista locazioni relativa a quella finestra.
Altrimenti una nuova finestra di help è aperta e la
lista locazioni per quella finestra viene impostata.
La lista locazioni per la finestra corrente non viene
in questo caso modificata.

```

```
*:exu* *:exusage*
```

```
:exu[sage] Mostra help relativo ai comandi Ex. Aggiunto per
simulare il comando Nvi.

```

```
*:viu* *:viusage*
```

```
:viu[sage] Mostra help relativo ai comandi in modo Normal.
Aggiunto per simulare il comando Nvi.

```

Se non si specifica alcun argomento per |:help| verrà aperto il file specificato dall'opzione 'helpfile'. Altrimenti si cerca il tag specificato in tutti i file "doc/tags" contenuti nelle directory specificate nell'opzione 'runtimepath'.

L'altezza iniziale della finestra di help può essere impostata nell'opzione 'helpheight' (default 20).

Andare a un argomento specifico usando i tag. Si può fare in due modi:

- Usare il comando "CTRL-]" mentre il cursore sta sul nome di un comando o di un'opzione. Questo può funzionare solo quando il tag è una chiave. "<C-Leftmouse>" e "g<LeftMouse>" possono essere usati invece che "CTRL-]".
- usare il comando ":ta {subject}". Questo funziona anche per parole che non siano chiavi.

Usare CTRL-T o CTRL-O per tornare indietro.
Usare ":q" per chiudere la finestra di help.

Se ci sono più corrispondenze per un dato argomento di ricerca, ecco come visualizzarle tutte, una dopo l'altra:

1. Aprire una finestra di help
2. Usare il comando ":tag" inserendo una "/" prima del tag. Per esempio: >

```
:tag /min
```
3. Usare ":tnext" per passare al prossimo tag trovato.

È possibile aggiungere file di help per plugin e altre funzionalità. Non è necessario per questo cambiare i file di help distribuiti con Vim. Vedere |add-local-help|.

Per preparare un file di help locale, vedere |write-local-help|.

Nota Le linee del titolo dei file di help locali sono aggiunte automaticamente

alla sezione "LOCAL ADDITIONS" del file di help "help.txt" |**local-additions**|. Questo viene fatto mentre il file viene visualizzato in Vim, il file su disco non è modificato. Ciò si ottiene leggendo tutti i file di help ed estraendo la prima linea di ogni file. I file in \$VIMRUNTIME/doc sono ignorati.

help-xterm-window

Se desiderate l'help in un'altra finestra xterm, potete usare questo comando: >

```
    :!xterm -e vim +help &
```

<

```

*:helpfind* *:helpf*
:helpf[ind]    Come |:help|, ma utilizza una finestra di dialogo per
                introdurre l'argomento. Presente solo per
                compatibilità all'indietro. Ora è eseguito nel menù
                ToolBar.FindHelp, invece che usare un dialogo
                autonomo. {solo se compilato con |+GUI_GTK|}

```

*:**helpt*** *:**helptags***
E154* **E150*** **E151*** **E152*** **E153*** **E670***

```

:helpt[ags] [++t] {dir}
                Genera il/i file delle tag di help tags per la
                directory {dir}.
                Quando {dir} vale ALL tutte le directory di nome "doc"
                contenute in 'runtimepath' saranno usate nella
                ricerca.

```

Tutti i file "*.txt" e "*.??x" nella directory e nelle eventuali sottodirectory sono esaminati per cercare una definizione di tag di help inclusa fra asterischi. I file "*.??x" contengono documentazione tradotta, e generano dei file di nome "tags-??", vedere |**help-translated**|. I file di tag così generati sono messi in ordine alfabetico. Se vengono trovate tag duplicate, viene dato un messaggio di errore. Un eventuale file di tag esistente viene sovrascritto senza dare alcun messaggio.

L'argomento facoltativo "++t" forza l'aggiunta del tag "help-tags". Questo viene anche fatto se la directory {dir} è \$VIMRUNTIME/doc.

Per ricostruire le tag di help nella directory di help correntemente in uso (il che richiede di poter scrivere in quella directory): >

```
    :helptags $VIMRUNTIME/doc
```

2. File di help tradotti

help-translated

È possibile aggiungere file di help tradotti, che si affiancano ai file di help originali in inglese. Vim ricerca file di help in tutte le directory "doc" elencate in 'runtimepath'. Questa possibilità è presente solo quando Vim è compilato con la funzionalità |+**multi_lang**|.

Ad oggi sono disponibili traduzioni per:

```

Cinese       - parecchi autori
Francese     - tradotto da David Blanchet
Italiano     - tradotto da Antonio Colombo e molti altri
Giapponese   - parecchi autori
Polacco      - tradotto da Mikolaj Machowski
Russo        - tradotto da Vassily Ragosin

```

Vedere il sito web di Vim per scaricarle: <http://www.vim.org/translations.php>

Un insieme di file tradotti consiste di questi file:

```

help.abx
howto.abx
...
tags-ab

```

"ab" è il codice di due lettere che indica il linguaggio. Quindi per l'italiano i nomi sono:

```
help.itx
howto.itx
...
tags-it
```

L'opzione `'helplang'` può essere impostata al/i linguaggio/i preferito/i. Il default è impostato a seconda dell'ambiente (informatico). Vim tenterà dapprima di trovare un tag corrispondente nel/i linguaggio/i preferito/i. L'inglese viene usato quando non ne venga trovato nessuno.

Per richiedere un tag in un dato linguaggio, aggiungere "@ab" a un tag, dove "ab" è il codice di due lettere del linguaggio. Esempio: >

```
:he user-manual@it
:he user-manual@en
```

La prima invocazione trova il manuale utente in italiano, anche senza che `'helplang'` sia stato impostato.

La seconda invocazione trova il manuale utente in inglese, anche qualora `'helplang'` sia stato impostato a "it".

Se si usa il completamento della linea dei comandi per il comando `:"help"`, l'estensione "@en" viene visualizzata solo quando il tag esista in più di un linguaggio. Se il tag esiste solo in lingua inglese, "@en" è omissso. Quando il primo elemento trovato ha un'estensione "@ab", e questa corrisponde al primo linguaggio elencato in `'helplang'`, anche "@ab" è omissso.

Se si usa `|CTRL-|` o `:"help!"` in un file di help non in inglese, Vim cercherà di trovare il tag nello stesso linguaggio. Se questo non viene trovato, Vim usa `'helplang'` per scegliere un linguaggio.

Ogni file di help deve usare codifiche latin1 o utf-8. Vim determina che la codifica è utf-8 se trova caratteri non-ASCII nella prima linea [di un file di help]. Per questo è necessario aver tradotto la linea di intestazione che contiene la dicitura "For Vim version".

La stessa codifica deve essere usata per tutti i file di help in una data lingua contenuti in una directory. Si possono usare codifiche differenti per linguaggi differenti e usare codifiche differenti per file di help di un dato linguaggio, a condizione che questi risiedano in directory differenti.

Suggerimenti per traduttori:

- Non traducete i tag. Ciò rende possibile usare `'helplang'` per specificare il linguaggio preferito. Potete aggiungere nuovi tag nella vostra lingua.
- Quando una parte di un file non è tradotta, aggiungere tag che richiamano la versione inglese, usando la notazione "tag@en".
- Preparete un pacchetto che contenga tutti i file e i tag e mettetelo a disposizione per lo scaricamento. Gli utenti possono scaricarlo in una delle directory "doc" e cominciare a usarlo. Informate di questo Bram, in modo che possa aggiungere un puntatore al vostro pacchetto su www.vim.org.
- Usate il comando `:"helptags|` per generare il file delle tag. È in grado di gestire tutti i linguaggi nella directory che gli specificate.

3. Scrittura file di help

help-writing

Per facilitarne l'uso, un file di help Vim di un plugin dovrebbe seguire il formato standard dei file di help di Vim. Se state scrivendo un nuovo file di help è meglio iniziare copiando uno dei file esistenti ed usandolo come modello.

La prima riga in un file di help dovrebbe avere il seguente formato:

```
*nome_file_help.txt*      Per Vim version 7.3      Ultima modifica: 2010 Jun 4
```

Il primo campo stabilisce il collegamento con il nome del file di help. Il secondo campo descrive la versione di Vim alla quale il file si applica. L'ultimo campo specifica la data di ultima modifica del file. Ogni campo è separato dall'altro da un carattere <Tab>.

In fondo al file di help, mettere una modeline di Vim per impostare le opzioni

'textwidth' e 'tabstop' e il 'filetype' a "help". Non impostare mai un'opzione globale in questa modeline, la cosa può avere conseguenze spiacevoli per chiunque legga quel file di help.

TAG

Per definire un tag di help, metterne il nome fra asterischi (*nome-tag*). nome-tag dovrebbe essere differente da tutti gli altri nomi di tag Vim e idealmente dovrebbe cominciare con il nome del plugin Vim. Il nome del tag è normalmente allineato a destra nella riga.

Per inserire un riferimento a un tag di help e per creare un legame diretto (hot-link), metterne il nome fra due barre (|) , per esempio |**help-writing**|.

Per inserire un riferimento a un comando Vim e creare un legame diretto (hot-link) racchiudete il nome fra due apici inversi, per esempio all'interno di `:filetype`. Questo costrutto sarà evidenziato come un comando, analogamente a un blocco di codice (vedere più sotto),

Quando si fa riferimento a un'opzione Vim nel file di help, mettere il nome dell'opzione fra due apici, per esempio 'statusline'.

EVIDENZIAMENTO

Per definire una intestazione mettere un carattere tilde (~) alla fine della linea. Questo farà sì che l'intestazione sia visualizzata in un colore differente. Per esempio

Intestazione~

Per separare sezioni in un file di help, inserite una serie di caratteri '=' in una riga, a partire dalla prima colonna. La linea di separazione della sezione sarà evidenziata in maniera differente.

Per inserire una serie di comandi "ex" parola per parola, mettere un carattere "maggiore di" (>) alla fine della linea che precede la serie di comandi, e un carattere "minore di" sulla prima linea non bianca che segue la serie di comandi. Ogni linea che inizia a colonna 1 è comunque sufficiente per marcare implicitamente la fine della serie di comandi di "ex". Per esempio: >

```
function Esempio_Funz()
  echo "Esempio"
endfunction
```

<

Quanto segue viene evidenziato in maniera differente in un file di help Vim:

- un tasto speciale, scritto sia con la notazione <> come per esempio <PageDown>, o come una sequenza di tasti includente CTRL, come per esempio CTRL-X
- qualsiasi parola racchiusa fra {graffe}, per esempio {lhs} e {rhs}

Le parole "Nota", "Note" e simili ricevono automaticamente un'evidenziazione particolare. Lo stesso vale per:

```
*Todo    qualcosa da fare
*Error   qualcosa di sbagliato
```

Maggiori dettagli si possono trovare in \$VIMRUNTIME/syntax/help.vim

```
vim:tw=78:ts=8:noet:ft=help:norl:
```

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