options.txt Per **Vim version 8.0.** Ultima modifica: 2018 May 15

VIM Manuale di Riferimento di Bram Moolenaar
Traduzione di questo testo: Autori Vari

Opzioni *options*

Per una breve spiegazione di ogni opzione vedere quickref.txt |**option-list**|.

Vim ha molte variabili interne anche di tipo binario che si possono impostare per ottenere effetti "speciali". Queste opzioni sono di tre tipi:
boolean *toggle*

booleana	può essere solo attiva o inattiva ("on"/"off")
numerica	ha un valore che è un numero
stringa	ha un valore che è una stringa di caratteri

==========

1. Impostare opzioni *set-option* *E764*

:se *:set*

:se[t] Lista ogni opzione di valore diverso dal default.

:se[t] all Lista tutte le opzioni, tranne quelle del terminale.

:se[t] termcap Lista tutte le opzioni del terminale. Se si sta usando una GUI i codici dei tasti (key-code) non sono listati, perché vengono generati internamente e non sono modificabili. Cambiare i valori dei tasti della GUI sarebbe comunque inutile...

E518 *E519*

:se[t] {opzione}? Lista il valore di {opzione}.

:se[t] {opzione} Opzione booleana: imposta, assegna valore "on"
Opzione numerica: lista il valore.
Opzione stringa: lista il valore.

:set-! *:set-inv*

:se[t] no{opzione} Opzione booleana: annulla, assegna valore "off"

:se[t] {opzione}! oppure
:se[t] inv{opzione} Opzione booleana: inverte il valore. {non in Vi}

:set-default *:set-&* *:set-&vi* *:set-&vim*

:se[t] {opzione}& Imposta opzione al suo valore di default, il quale può essere dipendente dal valore dell'opzione 'compatible'. {non in Vi}
:se[t] {opzione}&vi Imposta opzione al suo valore di default in Vi. {non in Vi}
:se[t] {opzione}&vim Imposta opzione al suo valore di default in Vim. {non in Vi}

:se[t] all& Imposta tutte le opzioni, al loro valore di default. I valori di queste opzioni non sono modificati:
- tutte le opzioni di terminali che iniziano con t_
- 'columns'
- 'cryptmethod'
- 'encoding'
- 'key'
- 'lines'
- 'term'
- 'ttymouse'
- 'ttytype'

Attenzione: Ci possono essere molti effetti laterali. {non in Vi}

:set-args *E487* *E521*

```
:se[t] {opzione}={valore}              o
:se[t] {opzione}:{valore}
                        Imposta opzione stringa o numerica a {valore}.
                        Per opzioni numeriche il valore può essere dato come
                        numero decimale, esadecimale (prefisso 0x) od ottale
                        (prefisso '0').
                        Il valore corrente si può inserire usando il carattere
                        'wildchar' (default: <Tab> oppure CTRL-E, laddove sia
                        attiva l'opzione 'compatible').
                        Vedere |cmdline-completion|.
                        Spazi bianchi fra {opzione} e '=' sono ammessi, ed
                        ignorati.  Spazi bianchi fra '=' e {valore} non sono
                        ammessi.
                        Vedere |option-backslash| per inserire spazi e
                        backslash in {valore}.

:se[t] {opzione}+={valore}                                      *:set+=*
                        Aggiunge {valore} a un'opzione numerica, o accoda
                        {valore} a un'opzione di tipo stringa.  Se l'opzione è
                        un elenco separato da virgole, una virgola viene
                        aggiunta, quando il valore preesistente non è una
                        stringa nulla.
                        Se l'opzione è una lista di flag, flag superflue
                        sono rimosse.  Quando si aggiunge un flag che è già
                        presente, il valore dell'opzione non cambia.
<                       Vedere anche |:set-args| più sopra.
                        {non in Vi}

:se[t] {opzione}^={valore}                                      *:set^=*
                        Moltiplica per {valore} un'opzione numerica, o
                        premette {valore} a un'opzione di tipo stringa.
                        Se l'opzione è un elenco separato da virgole, una
                        virgola viene aggiunta, laddove il valore preesistente
                        non sia una stringa nulla.
                        Vedere anche |:set-args| più sopra.
                        {non in Vi}

:se[t] {opzione}-={valore}                                      *:set-=*
                        Sottrae il {valore} da un'opzione numerica, o rimuove
                        il {valore} da un'opzione di tipo stringa, se è
                        presente.
                        Se il {valore} non è trovato in un'opzione di tipo
                        stringa, non c'è segnalazione di errore o
                        avvertimento.  Se l'opzione è un elenco separato da
                        virgole, anche una virgola viene tolta, quando
                        l'opzione non si riduce a una stringa nulla.
                        Quando l'opzione è una lista di flag, {valore} deve
                        essere esattamente nello stesso ordine in cui i flag
                        compaiono nell'opzione.  Per evitare problemi,
                        rimuovere i flag uno per volta.
                        Vedere anche |:set-args| più sopra.
                        {non in Vi}

Gli argomenti {opzione} di ":set" possono essere ripetuti.  Ad es.: >
        :set ai nosi sw=3 ts=3
Se uno degli argomenti risulta essere errato, viene visualizzato un messaggio
di errore, e gli argomenti seguenti sono ignorati.

                                                        *:set-verbose*
Quando 'verbose' è maggiore di zero, visualizzando il valore di un'opzione
viene detto anche dove è stata impostata l'ultima volta.  Ad es.: >
<       :verbose set shiftwidth cindent?
<         shiftwidth=4 ~
                Impostata l'ultima volta da modeline ~
          cindent ~
                Impostata l'ultima volta da ~
                     /usr/local/share/vim/vim60/ftplugin/c.vim ~
Ciò vale solo quando si chiede il valore di precise opzioni, non per
":verbose :set all" o ":verbose :set" senza argomenti.

Alcuni testi particolari:
        Impostata l'ultima volta da modeline ~
                L'opzione era stata impostata in una |modeline|.
```

        Impostata l'ultima volta da --cmd argomento ~
                L'opzione era stata impostata da un argomento fornito nella
                riga di comando |**--cmd**| o +.
        Impostata l'ultima volta da -c argomento ~
                L'opzione era stata impostata da un argomento fornito nella
                riga di comando |**-c**|, +, |**-S**| o |**-q**|.
        Impostata l'ultima volta da variabile d'ambiente ~
                L'opzione era stata impostata da una variabile d'ambiente,
                $VIMINIT, $GVIMINIT o $EXINIT.
        Last set from error handler ~
        Impostata l'ultima volta da gestore di errore ~
                L'opzione era stata annullata perché la sua valutazione
                aveva generato un errore.

Quando l'opzione è stata impostata direttamente dall'utente, non viene dato alcun messaggio "Impostata l'ultima volta". C'è un solo valore per tutte le opzioni locali aventi lo stesso nome. Quindi il messaggio vale per l'opzione con quel nome, e non necessariamente per il valore che l'opzione stessa ha in quel momento. [Ossia un'opzione locale valida in un altro buffer potrebbe essere stata specificata DOPO la stessa opzione per il buffer corrente. Di quest'ultima opzione in questo caso verrebbe dato il valore, ma non dove è stata impostata - NdT]
Quando l'opzione è stata impostata durante l'esecuzione di una funzione, di un comando utente, o da un autocomando, viene listato il nome dello script in cui è stata definita.
Nota Un'opzione può risultare impostata come effetto collaterale dell'aver impostato l'opzione 'compatible'.
{non disponibile se compilato senza la funzionalità |**+eval**|}

:set-termcap *E522*
{opzione} si può specificare nella forma "t_xx" per impostare un'opzione del terminale. Questa prevarrà sul valore corrente fornito da 'termcap'. Ciò può essere poi usato in una mappatura. Se la parte "xx" contiene caratteri speciali, usare la forma <t_xx>: >
        :set <t_#4>=^[Ot
In questo modo si può anche specificare un codice speciale prodotto da una sequenza di tasti normale. Ad es., se Alt-b significa <Esc>b, si usi: >
        :set <M-b>=^[b
(il "^[" qui sopra è un vero <Esc>, usare CTRL-V <Esc> per immetterlo)
Il vantaggio rispetto a una mappatura è che funziona in ogni situazione.

Potete definire ogni codice di tasti (key code), per es.: >
        :set t_xy=^[foo;
Anche se si usa un nome che non è riconosciuto, non viene emesso alcun messaggio. Si possono mappare questi codici come si preferisce: >
        :map <t_xy> qualcosa
<
E846
Se un codice di tasto non è impostato, è come se non esistesse. Se si chiede di visualizzarne il valore, si ottiene un messaggio di errore: >
        :set t_kb=
        :set t_kb
        E846: Key code not set: t_kb

Le opzioni t_xx non possono venire impostate da |**modeline**| o nel |**sandbox**|, per motivi di sicurezza.

La lista prodotta da ":set" ha un aspetto differente da Vi. Le opzioni stringa lunghe sono messe alla fine della lista. Il numero di opzioni è piuttosto elevato. L'output di "set all" probabilmente non sta in una schermata, e in questo caso quindi Vim richiede un input per proseguire |**more-prompt**|.

option-backslash
Per includere uno spazio nel valore di un'opzione di tipo stringa occorre proteggerlo con un backslash [\]. Per includere un backslash, ne vanno specificati due. Ciò implica che il numero di backslash nel valore di un'opzione è la metà di quelli effettivamente specificati (con eventuale arrotondamento all'unità inferiore):
Alcuni esempi: >
   :set tags=tags\ /usr/tags   produce "tags /usr/tags"
   :set tags=tags\\,file   produce "tags\,file"
   :set tags=tags\\\ file   produce "tags\ file"

Il carattere "|" separa un comando ":set" da un comando successivo. Per includere un "|" nel valore dell'opzione, usare "\|". Questo esempio imposta l'opzione 'titlestring' a "Buon|Giorno": >
```
:set titlestring=Buon\|Giorno
```
Così invece si imposta l'opzione 'titlestring' a "Buon" e 'iconstring' a "Giorno": >
```
:set titlestring=Buon|set iconstring=Giorno
```

Analogamente, il carattere doppio apice (") indica l'inizio di un commento. Per includere il '"' come valore di un'opzione, usare invece '\"'. Questo esempio imposta l'opzione 'titlestring' a 'Buon "Giorno"': >
```
:set titlestring=Buon\ \"Giorno\"
```

Per MS-DOS e WIN32 i backslash nei nomi di file sono per lo più conservati. Più precisamente: per opzioni che si aspettano un nome di file (quelle in cui a variabili d'ambiente viene attribuito il loro valore), un backslash che precede un carattere normale nel nome del file non viene rimosso. Ma un backslash che precede un carattere speciale (spazio, backslash, virgola, etc.) è usato come spiegato più sopra.
C'è un caso speciale, quello in cui il valore comincia con "\\": >
```
:set dir=\\machine\path          produce "\\machine\path"
:set dir=\\\\machine\\path       produce "\\machine\path"
:set dir=\\path\\file            produce "\\path\file" (errato!)
```
Nel primo esempio sopra il "\\" iniziale è mantenuto, ma nel secondo, i "\" sono dimezzati. Ciò assicura che l'assegnamento funzioni, sia che ci si aspetti un dimezzamento dei "\", sia che ci si aspetti che vengano mantenuti. Il terzo esempio dà un risultato che non è probabilmente quello voluto. Da evitare.

add-option-flags *remove-option-flags* *E539* *E550* *E551* *E552*

Alcune opzioni sono liste di flag. Quando volete aggiungere un flag a una di tali opzioni, senza modificare quelle già presenti, potete fare così: >
```
:set guioptions+=a
```
E per levare un flag da un'opzione così: >
```
:set guioptions-=a
```
Ciò rimuove il flag 'a' da 'guioptions'.
Nota Dovreste aggiungere o levare un flag alla volta. Se 'guioptions' ha il valore "ab", usare "set guioptions-=ba" non funziona, poiché la stringa "ba" non viene trovata.

:set_env *expand-env* *expand-environment-var*

Variabili d'ambiente in opzioni di tipo stringa vengono valuatte in alcuni casi specifici. Se la variabile d'ambiente esiste, il '$' e il successivo nome di variabile d'ambiente sono rimpiazzati dal valore della variabile. Se invece la variabile d'ambiente non è definita, il '$' e il nome non vengono modificati. Ogni carattere non usato in un nome (ovvero diverso da una lettera, cifra o "_") può essere messo dopo il nome della variabile di ambiente. Quel carattere e quelli che lo seguono sono aggiunti al valore della variabile d'ambiente. Ad es.: >
```
:set term=$TERM.new
:set path=/usr/$INCLUDE,$HOME/include,.
```
Quando si aggiunge o toglie una stringa da un'opzione con ":set opt+=valore" o ":set opt-=valore" la valutazione è fatta prima di aggiungere o togliere.

Gestione di opzioni locali *local-options*

Alcune opzioni valgono solo in una finestra o buffer. Ogni finestra o buffer ha la propria copia di quest'opzione, e quindi ognuna può assumere un valore distinto. Ciò consente di impostare 'list' in una finestra, e non in un'altra. E di impostare 'shiftwidth' a 3 in un buffer e a 4 in un altro.

Quanto segue spiega cosa succede a queste opzioni locali in situazioni particolari. Non è realmente necessario conoscere ogni dettaglio, poiché Vim per lo più usa i valori di opzione che vi aspettate. Sfortunatamente, fare quello che l'utente si aspetta è un po' complicato...

Dividendo in due una finestra, le opzioni locali sono copiate nella nuova finestra. Subito dopo la divisione, quindi, i contenuti delle due finestre sono molto simili.

Quando si apre un nuovo buffer, i valori delle sue opzioni locali devono

essere inizializzati. Poiché le opzioni del buffer corrente [il "vecchio" buffer - NdT] potrebbero essere state impostate solo per quel buffer, esse vengono ignorate. Viene invece usato il valore impostato per l'opzione globale corrispondente all'opzione locale al buffer e che viene usata per i nuovi buffer. Col comando ":set" vengono cambiati sia il valore locale che quello globale. Col comando ":setlocal" solo il valore locale viene cambiato, ma questo valore non è usato quando di apre un nuovo buffer.

Quando si modifica un buffer che era già stato aperto in precedenza, le opzioni relative alla finestra che è stata chiusa per ultima sono usate ancora. Se il buffer in questione era già stato editato in questa finestra, i valori usati in precedenza vengono riutilizzati. Altrimenti vengono utilizzati i valori dall'ultima finestra chiusa in cui il buffer era stato editato l'ultima volta.

È possibile impostare un'opzione locale alla finestra a seconda del tipo di buffer. Se editate un altro buffer nella stessa finestra, non volete continuare ad usare le stesse opzioni locali della finestra. Per questo Vim mantiene un valore globale per ogni opzione locale alla finestra, e questo viene usato quando si passa ad editare un altro buffer. Ogni finestra ha una propria copia di questi valori. Questi ultimi sono locali per questa finestra, ma globali per tutti i buffer che si editano nella finestra. Così potete fare: >

```
:e uno
:set list
:e due
```

Così l'opzione 'list' resterà impostata anche in "due", poiché con il comando ":set list" avete impostato anche il valore globale. >

```
:set nolist
:e uno
:setlocal list
:e due
```

Così l'opzione 'list' non è impostata, perché ":set nolist" imposta il valore globale, ":setlocal list" cambia solo il valore locale e ":e two" utilizza il valore globale. Nota Se in seguito battete: >

```
:e uno
```

Ritroverete impostato il valore 'list' che era attivo l'ultima volta che avete editato "uno". Le opzioni locali di una finestra sono ricordate per ogni buffer. Questo rimane vero anche quenado il buffer non è caricato, ma esse vanno perse quando il buffer è cancellato in maniera definitiva |**:bwipe**|.

:setl *:setlocal*

:setl[ocal] ... Come ":set" ma imposta solo il valore localmente nella finestra o buffer corrente. Non tutte le opzioni hanno un valore locale. Se l'opzione non ha un valore locale, viene impostato quello globale.
Con l'argomento "all": visualizza tutti i valori di tutte le opzioni locali.
Senza argomenti: visualizza i valori locali, diversi dal valore di default, per tutte le opzioni locali.
Nella visualizzazione di una specifica opzione locale, viene visualizzato il valore locale. Per un'opzione booleana globale/locale, quando è in uso il valore globale, "--" è visualizzato prima del nome dell'opzione.
Per un'opzione globale, viene visualizzato il valore globale (ma questo potrebbe cambiare in futuro).
{non in Vi}

:setl[ocal] {opzione}< Imposta il valore locale di {opzione} al valore globale corrispondente, facendone una copia.
{non in Vi}

:se[t] {option}< Per opzioni |**global-local**|: Elimina il valore locale di {opzione}, in modo da utilizzare quello globale.
{non in Vi}

:setg *:setglobal*

:setg[lobal] ... Come ":set" ma imposta solo il valore globale di una opzione locale senza modificare il valore locale.
Nella visualizzazione di un'opzione viene visualizzato il valore globale.
Con l'argomento "all": visualizza tutti i valori

globali di tutte le opzioni locali.
Senza argomenti: visualizza i valori globali, diversi dal valore di default, per tutte le opzioni locali.
{non in Vi}

Per opzioni locali al buffer e alla finestra:

Comando	valore globale	valore locale ~
:set option=valore	imposta	imposta
:setlocal option=valore	-	imposta
:setglobal option=valore	imposta	-
:set option?	-	visualizza
:setlocal option?	-	visualizza
:setglobal option?	visualizza	-

Opzioni globali con un valore locale *global-local*

Le opzioni sono globali quando si usa per lo più un valore solo per tutti i buffer e le finestre. Per alcune opzioni globali è utile poter usare talora un valore locale differente. Potete impostare il valore locale con ":setlocal". Quel buffer o finestra utilizzerà il valore locale, mentre gli altri buffer e finestre continueranno a usare il valore globale.

Ad es., potreste avere due finestre, entrambe con un codice sorgente C. Queste usano l'opzione globale 'makeprg'. Se scrivete in una delle due finestre: >
```
:set makeprg=gmake
```
anche l'altra finestra passerà allo stesso valore. Non serve impostare l'opzione 'makeprg' anche nell'altra finestra di codice sorgente C. Peraltro, se cominciare ad editare un file Perl in una nuova finestra, vorreste usare un altro 'makeprg' per questo, senza modificare il valore usato per i file di codice sorgente C. Usate questo comando: >
```
:setlocal makeprg=perlmake
```
Potete ritornare ad usare il valore globale annullando quello locale: >
```
:setlocal makeprg=
```
Ciò funziona solo per un'opzione di tipo stringa. Per un'opzione booleana dovete usare il flag "<", come in quest'esempio: >
```
:setlocal autoread<
```
Nota Per opzioni non booleane usare "<" copia il valore globale al valore locale, senza tornare ad usare il valore globale (ciò è rilevante nel momento in cui il valore globale venga cambiato in seguito). Si può anche usare: >
```
:set path<
```
Ciò renderà nullo il valore locale di 'path', in modo da usare il valore globale. Quindi ciò equivale a dare il comando: >
```
:setlocal path=
```
Nota: In futuro ulteriori opzioni globali potrebbero divenire globali-locali. L'uso di ":setlocal" per un'opzione globale potrebbe allora essere modificato.

Impostare 'filetype' (tipo file)

:setf[iletype] [FALLBACK] {filetype} *:setf* *:setfiletype*
Impostare l'opzione 'filetype' a {tipo_file}, ma solo se questo non è già stato fatto da una serie di autocomandi (incassati).
Una abbreviazione per: >
```
:if !did_filetype()
:  setlocal filetype={tipo_file}
:endif
```
<
Questo comando è usato in un file filetype.vim per evitare di impostare due volte l'opzione 'filetype' e far caricare così differenti impostazioni e file di colorazione sintattica.

Quando si specifica l'argomento opzionale FALLBACK, un successivo comando :setfiletype prevarrà su quello specificato tramite 'filetype'. Questo argomento va utilizzato qualora la determinazione del tipo di file sia stata fatta tirando a indovinare.
|**did_filetype()**| risponderà "false" dopo questo comando.

{non in Vi}

option-window *optwin*
:bro[wse] se[t] *:set-browse* *:browse-set* *:opt* *:options*
:opt[ions] Apre una finestra per vedere e impostare tutte le opzioni. Le opzioni sono raggruppate per funzione. Un breve aiuto è visualizzato per ogni opzione. Dando <INVIO> sul testo di aiuto breve si apre una nuova finestra, con ulteriore aiuto per l'opzione. Modificare il valore dell'opzione e dare <INVIO> sulla riga "set" per impostare il nuovo valore. Per opzioni locali a un buffer o a una finestra l'ultima finestra acceduta è quella usata per impostare il valore dell'opzione, a meno che non si tratti di una finestra di aiuto, nel qual caso si usa la finestra che sta sotto a quella di aiuto (non contando la finestra delle opzioni (di nome "option-window")). {non disponibile se compilato senza la funzionalità |**+eval**|}

$HOME
Usare "~" equivale a usare la variabile d'ambiente "$HOME", ma è possibile solo all'inizio di un'opzione, o dopo uno spazio o una virgola.

Su sistemi Unix si può usare anche "~utente". È rimpiazzato dal nome della home directory dell'utente "utente". Ad es.: >
```
:set path=~mool/include,/usr/include,.
```

Su sistemi Unix si può usare anche la forma "${HOME}". Il nome fra {} può contenere caratteri diversi da quelli normalmente usati in un nome (ossia diversi da lettere, cifre e "_"). Nota Se volete usare queste variabili nel comando "gf", i caratteri '{' e '}' vanno aggiunti a 'isfname'.

NOTA: La valutazione di variabili d'ambiente e di "~/" viene effettuata al momento dell'esecuzione del comando ":set", non quando si assegna un valore a un'opzione con ":let".

$HOME-windows
In MS-Windows, se $HOME non è definita come variabile d'ambiente, Vim la imposta quando viene richiamato, attribuendole il valore della valutazione di $HOMEDRIVE$HOMEPATH.
Se $HOMEDRIVE non è impostato, si usa $USERPROFILE.

Questo valore valutato non è reso disponibile nelle variabili d'ambiente. Ciò è importante quando si esegua un comando fuori da Vim: >
```
:echo system('set | findstr ^HOME=')
```
e >
```
:echo luaeval('os.getenv("HOME")')
```
non dovrebbero visualizzare nulla (cioè la stringa vuota) anche se exists('$HOME') dà come risultato "true" ("vero").
Se si imposta $HOME a una stringa diversa dalla stringa nulla, questo valore verrà reso disponibile ai sottoprocessi.

Nota La lunghezza macchina di un'opzione valutata è limitato. Il limite dipende dal sistema, per lo più è qualcosa come 256 o 1024 caratteri.

:fix *:fixdel*
:fix[del] Imposta il valore di 't_kD':

't_kb' è	't_kD' diventa
CTRL-?	CTRL-H
non CTRL-?	CTRL-?

(CTRL-? è 0177 in ottale, 0x7f in esadecimale)
{non in Vi}

Se il codice terminale del vostro tasto <Cancella> è impostato male, ma il codice per il backspace è corretto, potete inserire nel vostro file .vimrc: >
```
:fixdel
```
< Ciò funziona a prescindere da quale sia il codice terminale per il tasto backspace.

Se il codice terminale del vostro tasto di backspace

è impostato male, usate: >
```
:if &term == "tipo_terminale"
:  set t_kb=^V<BS>
:  fixdel
:endif
```
< Dove "^V" è CTRL-V e "<BS>" è il tasto di backspace (non immettete quattro caratteri!). Sostituite a "tipo_terminale" il nome del vostro terminale.

Se il vostro tasto <Cancella> spedisce una sequenza strana di chiavi (diversa da CTRL-? o CTRL-H) non potete usare ":fixdel". Usate invece: >
```
:if &term == "tipo_terminale"
:  set t_kD=^V<Delete>
:endif
```
< Dove "^V" è CTRL-V e "<Delete>" è il tasto <Cancella> (non immettete otto caratteri!). Sostituite a "tipo_terminale" il nome del vostro terminale.

Linux-backspace

Nota Per Linux: Come valore di default il tasto backspace invia CTRL-?, il che è errato. Potete correggerlo mettendo questa riga nel vostro file rc.local: >
```
echo "keycode 14 = BackSpace" | loadkeys
```
<

NetBSD-backspace

Nota Per NetBSD: Se il vostro backspace non produce il carattere giusto, tentate: >
```
xmodmap -e "keycode 22 = BackSpace"
```
< Se questo risolve, mettere nel vostro file .Xmodmap: >
```
keysym 22 = BackSpace
```
< Dovete ripartire perché la modifica diventi effettiva.

==
2. Impostazione automatica delle opzioni *auto-setting*

Oltre a modificare le opzioni con il comando ":set", ci sono tre alternative per impostarle automaticamente per uno o più file:

1. Quando Vim viene avviato le inizializzazioni vengono lette in diversi posti. Vedere |**initialization**|. Molte di queste vengono impiegate in tutte le sessioni di lavoro, ed alcune di queste dipendono dalla directory da cui Vim viene avviato.
   Potete creare un file di inizializzazione con |**:mkvimrc**|, |**:mkview**| e |**:mksession**|.
2. Se iniziate a lavorare con un file nuovo, vengono eseguiti i comandi automatici.
   Ciò può venire usato per impostare opzioni per dei file corrispondenti ad un modello particolare e molte altre cose. Vedere |**autocommand**|.
3. Se iniziate a lavorare con un file nuovo, e l'opzione 'modeline' è attiva, un certo numero di righe all'inizio ed alla fine del file verranno controllate per modeline. Ciò viene spiegato qui.

modeline *vim:* *vi:* *ex:* *E520*

Ci sono due forme di modeline. La prima:
```
[text]{white}{vi:|vim:|ex:}[white]{options}
```

[text]	qualsiasi testo o nulla
{white}	almeno un carattere bianco (<Space> o <Tab>)
{vi:|vim:|ex:}	la stringa "vi:", "vim:" o "ex:"
[white]	spazio bianco facoltativo
{options}	un elenco di impostazioni di opzioni, separate da uno spazio bianco o da un ':', dove ogni parte tra ':' è l'argomento di un comando ":set" (può essere nullo)

Esempi:
```
vi:noai:sw=3 ts=6 ~
vim: tw=77 ~
```

La seconda forma (compatibile con alcune versioni di Vi):

```
        [text]{white}{vi:|vim:|Vim:|ex:}[white]se[t] {options}:[text]

[text]                  qualsiasi testo o nulla
{white}                 almeno un carattere bianco (<Space> o <Tab>)
{vi:|vim:|Vim:|ex:}     la stringa "vi:", "vim:", "Vim:" o "ex:"
[white]                 spazio bianco facoltativo
se[t]                   la stringa "set " o "se " (notare lo spazio); quando
                        si usa "Vim" deve essere scritta come "set".
{options}               un elenco di opzioni, separate da spazi bianchi, che
                        è l'argomento di un comando ":set"
:                       un segno di due punti
[text]                  qualsiasi testo o nulla
```

Esempi:

```
   /* vim: set ai tw=75: */ ~
   /* Vim: set ai tw=75: */ ~
```

Lo spazio bianco prima di {vi:|vim:|Vim:|ex:} è necessario. Ciò riduce al minimo la possibilità che una parola normale come "lex:" crei impiccio. C'è una sola eccezione: "vi:" e "vim:" possono trovarsi anche all'inizio della riga (per compatibilità con la versione 3.0). L'uso di "ex:" all'inizio della riga verrà ignorato (potrebbe essere l'abbreviazione di "example:").

modeline-local

Le opzioni vengono impostate con ":setlocal": Il nuovo valore verrà applicato soltanto al buffer che contiene la finestra. Sebbene sia possibile impostare opzioni globali da una modeline, ciò è insolito. Se aveste due finestre aperte ed i file entro di esse con la stessa opzione globale impostata a valori diversi, il risultato dipende da quale file sia stato aperto per ultimo.

modeline-version

Se la modeline fosse soltanto da usare in alcune versioni di Vim, il numero di versione può essere specificato dove "vim:" o "Vim:" viene usato:

```
        vim{vers}:      versione {vers} o successiva
        vim<{vers}:     versione precedente a {vers}
        vim={vers}:     versione {vers}
        vim>{vers}:     versione successiva a {vers}
```

{vers} vale 700 per Vim 7.0 (cento volte il numero maggiore di versione più il numero minore moltiplicato per 10).
Ad esempio, per usare una modeline soltanto per Vim 7.0 e successivi:

```
        /* vim700: set foldmethod=marker: */ ~
```

Per usare una modeline per Vim dopo la versione 7.2:

```
        /* vim>702: set cole=2: */ ~
```

Non ci possono essere spazi bianchi tra "vim" ed il ":".

Il numero di righe che viene verificato può venire impostato con l'opzione 'modelines'. Se 'modeline' fosse inattiva o 'modelines' avesse il valore di 0 nessuna riga verrebbe provata.

Notare che per la prima forma viene usato il resto della riga, così una riga come:

```
   /* vi:ts=4: */ ~
```

darà un messaggio di errore a causa dei "*/" trascinati. Questa riga va bene:

```
   /* vi:set ts=4: */ ~
```

Se venisse trovato un errore il resto della riga verrà saltato.

Se voleste includere un ':' entro un comando set fatelo precedere da un '\'. Il backslash davanti a ':' verrà rimosso. Esempio:

```
   /* vi:set dir=c\:\tmp: */ ~
```

Ciò imposta l'opzione 'dir' a "c:\tmp". Soltanto un unico backslash prima dei ':' viene rimosso. Così per inserire un "\:" dovete specificare "\\:".

Nessun comando oltre "set" è consentito, per motivi di sicurezza (qualcuno potrebbe creare un file di testo "cavallo di Troia" con modeline). E non tutte le opzioni possono essere impostate, Per alcune opzioni è impostato un flag, in modo che quando la si usa il |**sandbox**| ha effetto. Nonostante ciò c'è sempre un piccolo rischio che una modeline possa creare problemi. Ad es. quando qualcuno in vena di scherzare imposta 'textwidth' a 5, tutte le vostre righe sono visualizzate in maniera inaspettata. Per cui dovreste inibire le

modeline, prima di modificare del testo su cui avete dei dubbi.
L'ftplugin della posta, ad esempio, fa questo.

Suggerimento: Se voleste fare qualcos'altro dall'impostare un'opzione,
potreste definire un autocomando che cerchi entro il file una specifica
stringa. Ad esempio: >
```
au BufReadPost * if getline(1) =~ "VAR" | call SetVar() | endif
```
E definire la funzione SetVar() che fa qualcosa con la riga che contiene
"VAR".

==
3. Sommario delle opzioni *option-summary*

Nell'elenco seguente vengono citate tutte le opzioni con il loro nome intero e
con l'abbreviazione se ne esiste una. Si possono usare entrambe le forme.

In questo documento quando un'opzione booleana risulta "set" ciò significa che
è stato adoperato ":set option". Quando un'opzione risulta "reset", è stato
usato ":set nooption".

Per alcune opzioni ci sono due valori di default: Il "default Vim", che viene
usato quando 'compatible' non è impostato, ed il "default Vi", che si usa
quando è impostato 'compatible'.

La maggior parte delle opzioni è valida in tutte le finestre e buffer. Ce ne
sono poche che dipendono da come il testo viene presentato entro una finestra.
Queste possono venire impostate ad un valore diverso per ciascuna finestra.
Ad esempio l'opzione 'list' può venire impostata su di una finestra e non
esserlo in un'altra per il medesimo testo, dando entrambe le visualizzazioni
contemporaneamente. Ci sono poche opzioni applicabili solo a un certo file.
Queste possono avere valore diverso per ogni file o buffer. Ad esempio
l'opzione 'textwidth' può essere 78 per un file di testo normale e 0 per un
programma C.

globale	unica opzione per ogni buffer e finestra
locale alla finestra	ogni finestra ha la propria copia di questa opzione
locale al buffer	ogni buffer ha la propria copia di questa opzione

Creando una nuova finestra i valori delle opzioni della finestra attualmente
attiva vengono usati come valore di default nelle opzioni specifiche per la
finestra. Per le opzioni specifiche per il buffer ciò dipende dai flag 's'
e 'S' nell'opzione 'cpoptions'. Se in 's' vengono inseriti (questo è il
default) i valori per le opzioni del buffer, questi vengono copiati dal buffer
attualmente attivo quando il buffer è stato il primo a venire attivato.
Se 'S' è presente le opzioni vengono copiate ogni volta che il buffer venga
attivato, ciò è almeno come sono impostate le opzioni globali. Se 's' ed 'S'
non fossero presenti, le opzioni verrebbero copiate dal buffer correntemente
attivo quando il buffer viene creato.

Opzioni nascoste ("hidden") *hidden-options*

Non tutte le opzioni sono supportate in tutte le versioni. Ciò dipende dalle
funzionalità scelte al tempo della compilazione, e talora dal sistema
operativo in cui si lavora. Un'osservazione a questo proposito la si ritrova
tra parentesi graffe, nel resto di questo testo. Quando un'opzione non è
supportata, può essere impostata lo stesso, senza ottenere alcuna segnalazione
di errore, e in questo caso si parla di un'opzione nascosta. Non si può però
controllare il vale di un'opzione nascosta, perché non viene memorizzato.

Per controllare se l'opzione "foo" possa venire usata con ":set" usare
qualcosa come: >
```
if exists('&foo')
```
Questa funzione restituisce un valore di "true" ("vero") anche per un'opzione
nascosta. Per controllare se l'opzione "foo" sia davvero supportata usare
qualcosa come: >
```
if exists('+foo')
```
<
E355
Una tabella riassuntiva per le opzioni con una breve descrizione può venire
trovata presso |**Q_op**|.

'aleph' *'al'* *aleph* *Aleph*
'aleph' 'al' numero (default: 128 per MS-DOS, 224 altrimenti)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+rightleft|}

Il codice ASCII per le prime lettere dell'alfabeto ebraico. La routine che descrive la tastiera nel modo Ebraico, sia in modo Insert (quando hkmap è impostato) che nella riga di comando (battendo CTRL-_) emette i caratteri Ebraici nel campo [aleph..aleph+26]. aleph=128 viene applicato al codice PC, ed aleph=224 si applica ad ISO 8859-8. Vedere |**rileft.txt**|.

'allowrevins' *'ari'* *'noallowrevins'* *'noari'*
'allowrevins' 'ari' booleana (default: off)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+rightleft|}
Permette CTRL-_ in entrambi i modi Insert e Command-line. Questo di default viene escluso, per evitare che l'utente che per caso battesse CTRL-_ invece di SHIFT-_ vada in modo Insert da destra verso sinistra, e non sappia più come uscirne. Vedere 'revins'.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'altkeymap' *'akm'* *'noaltkeymap'* *'noakm'*
'altkeymap' 'akm' booleana (default: off)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+rightleft|}
Se impostato a on, la seconda lingua è il Farsi. In modo editing CTRL-_ fa passare la tastiera da Farsi ad Inglese, se 'allowrevins' è impostato.

Se impostato a off, la mappa di tastiera si alterna fra Ebraico ad Inglese. Ciò è utile per far partire Vim in modo nativo, cioè in Inglese (modo da sinistra a destra) ed avere come seconda lingua di default Farsi o Ebraico (modo da destra a sinistra).
Vedere |**farsi.txt**|.

'ambiwidth' *'ambw'*
'ambiwidth' 'ambw' stringa (default: "single")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+multi_byte|}
Funziona soltanto se 'encoding' è "utf-8" o altra codifica Unicode.
Dice a Vim cosa fare con i caratteri aventi la East Asian Width Class ambigua (come Euro, Registrato, Copyright, lettere greche, caratteri cirillici).

Ora ci sono due valori possibili:
"single": Usa la stessa larghezza dei caratteri come in US-ASCII. Ciò è quanto si aspetta la maggioranza degli utenti.
"double": Usa il doppio della larghezza dei caratteri ASCII.
E834 *E835*
Il valore "double" non si può usare se 'listchars' o 'fillchars' contengono un carattere che sarebbe di larghezza doppia.

Ci sono dei font CJK (Chinese/Japanese/Korean) per cui la larghezza dei glifi per questi caratteri è basata soltanto su quanti ottetti richiedono nelle codifiche CJK legacy/tradizionale. In queste codifiche i caratteri Euro, Marchio-Registrato, le lettere greche/cirilliche sono rappresentate da due ottetti, perciò questi font hanno glifi "wide" (larghi) per essi. Ciò è anche vero per alcuni caratteri che disegnano righe, usati per creare tabelle entro file di testo. Perciò, quando un font CJK viene usato nella GUI di Vim o Vim gira entro un terminale (emulatori) che usi font CJK (o Vim giri entro un "xterm" chiamato con l'opzione "-cjkwidth"), quest'opzione dovrebbe essere impostata su "double" per far

corrispondere l'ampiezza percepita da Vim con quella dei glifi nel font. Forse dovrà venire impostata a "double" sotto CJK Windows 9x/ME o Windows 2k/XP quando il "locale" [la lingua] di sistema sia impostato su uno dei "locale" di CJK. Vedere Unicode Standard Annex #11 (http://www.unicode.org/reports/tr11).

Vim può impostare quest'opzione automaticamente al momento dell'invocazione quando Vim è compilato con la funzionalità |**+termresponse**| e se |**t_u7**| è impostato alla sequenza di protezione che richiede di riferire la posizione del cursore. Il risultato può essere trovato nella variabile |**v:termu7resp**|.

'antialias' *'anti'* *'noantialias'* *'noanti'*
'antialias' 'anti' booleana (default: off)
globale
{non in Vi}
{disponibile soltanto se compilato con la GUI abilitata su Mac OS X}

quest'opzione è valida solo per la versione GUI di Vim su Mac OS X v10.2 o successivo. Se impostata, Vim userà font chiari ("antialiased"), che possono essere più facili da leggere in alcune dimensioni su alcuni monitor.
Impostare quest'opzione può talvolta causare problemi se 'guifont' fosse impostato al valore di default (stringa nulla).
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'autochdir' *'acd'* *'noautochdir'* *'noacd'*
'autochdir' 'acd' booleana (default: off)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità stessa, usare exists("+autochdir") per controllare}

Quando impostato, Vim cambia il proprio valore della directory di lavoro corrente ogni volta che aprite un file, cambiate buffer, cancellate un buffer o aprite/chiudete una finestra. Passerà alla directory che contiene il file che era stato aperto o selezionato.
Nota: Quando quest'opzione è attiva, alcuni plugin potrebbero non funzionare.

'arabic' *'arab'* *'noarabic'* *'noarab'*
'arabic' 'arab' booleana (default: off)
locale alla finestra
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+arabic**|}

Quest'opzione può venire impostata per scrivere testo in Arabo.
Impostare quest'opzione causerà:
- L'impostazione dell'opzione 'rightleft' se non è impostato 'termbidi'.
- L'impostazione dell'opzione 'arabicshape' se non è impostato 'termbidi'.
- L'impostazione dell'opzione 'keymap' ad "arabic"; in modo Insert CTRL-^ passa tra la scrittura in Inglese e la mappatura di tastiera Araba.
- L'impostazione dell'opzione 'delcombine'.

Notare che 'encoding' deve essere "utf-8" per lavorare con il testo Arabo.

Disabilitare quest'opzione causerà:
- Inattivazione dell'opzione 'rightleft'.
- Disabilitazione all'uso di 'keymap' (senza cambiarne il valore).

Notare che 'arabicshape' e 'delcombine' non vengono messi a off (si tratta di un'opzione globale).
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.
Vedere anche |**arabic.txt**|.

'arabicshape' *'arshape'*
'noarabicshape' *'noarshape'*
'arabicshape' 'arshape' booleana (default: on)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+arabic**|}

Se impostata e 'termbidi' è disabilitata, sono abilitate le correzioni visuali di carattere, necessarie per consentire la visualizzazione della lingua Araba. Lo "shaping", in sostanza, viene abilitato. Questo termine indica una serie di effetti, che comprendono:
a) il cambio o la trasformazione di caratteri basato sulla loro posizione entro una parola (iniziale, media, finale ed isolata).
b) l'abilitazione della capacità di comporre caratteri.
c) l'abilitazione della richiesta combinazione di alcuni caratteri.
Se disabilitata, la visualizzazione del carattere mostra la vera forma del carattere, preso a sé stante.
L'arabo è una lingua complessa che richiede altre impostazioni, per ulteriori dettagli vedere |**arabic.txt**|.
NOTA: Quest'opzione è impostata a on se si attiva 'compatible'.

'autoindent' *'ai'* *'noautoindent'* *'noai'*
'autoindent' 'ai' booleana (default: off)
locale al buffer
Copiare l'indentatura della riga corrente quando inizia una nuova riga (battendo <CR> in modo Insert o quando si usano i comandi "o" o "O"). Se non si scrive nulla sulla nuova riga, eccetto <BS> o CTRL-D e poi battete <Esc>, CTRL-O o <CR>, l'indentatura verrà nuovamente cancellata. Spostare il cursore su un'altra riga produce lo stesso effetto, a meno che in 'cpoptions' sia inclusa il flag "I". Quando l'autoindentatura è impostata, la formattazione (con il comando "gq" o quando si arriva 'textwidth' in modo Insert) usa l'indentatura della prima riga.
Quando sono impostate 'smartindent' o 'cindent' l'indentatura viene cambiata in modo diverso.
L'opzione 'autoindent' viene messa a off quando l'opzione 'paste' è impostata e ripristinata quando 'paste' è messa a off.
{piccola differenza con Vi: Dopo che sia stata cancellata l'indentatura quando battete <Esc> o <CR>, la posizione del cursore quando vi spostate verso l'alto o verso il basso è dopo l'indentatura cancellata; Vi pone il cursore da qualche parte nell'indentatura cancellata}.

'autoread' *'ar'* *'noautoread'* *'noar'*
'autoread' 'ar' booleana (default: off)
globale o locale al buffer |**global-local**|
{non in Vi}
Quando scopre che un file è stato modificato al di fuori di Vim e non entro Vim, lo rilegge automaticamente.
Se il file fosse stato cancellato ciò non verrebbe fatto. |**timestamp**|
Se quest'opzione ha un valore locale, usate questo comando per ritornare all'uso del valore globale: >
:set autoread<
<

'autowrite' *'aw'* *'noautowrite'* *'noaw'*
'autowrite' 'aw' booleana (default: off)
globale
Scrive i contenuti di un file, se è stato modificato, a ciascun comando :next, :rewind, :last, :first, :previous, :stop, :suspend, :tag, :!, :make, CTRL-] e CTRL-^; e quando un comando :buffer, CTRL-O, CTRL-I, '{A-Z0-9}, o `{A-Z0-9} va in modifica su un altro file.
Notare che per alcuni comandi l'opzione 'autowrite' non viene usata, vedere 'autowriteall' per questo.

'autowriteall' *'awa'* *'noautowriteall'* *'noawa'*
'autowriteall' 'awa' booleana (default: off)
globale
{non in Vi}
Come 'autowrite', ma anche usato per i comandi ":edit", ":enew", ":quit", ":qall", ":exit", ":xit", ":recover" e nel chiudere la finestra di Vim.
Impostare quest'opzione implica anche che Vim si comporti come se fosse stata attivata l'opzione 'autowrite'.

'background' *'bg'*
'background' 'bg' stringa (default: "dark" o "light", vedi sotto)
globale
{non in Vi}
Quando impostato a "dark", Vim proverà ad usare i colori che appaiono meglio su di uno sfondo scuro. Quando impostato su "light", Vim

tenterà di usare i colori che appaiono meglio su uno sfondo chiaro.
Ogni altro valore non è consentito.
Vim prova ad impostare il valore di default a seconda del terminale
usato.
Ciò non sempre produce buoni risultati.
Impostare quest'opzione non cambia il colore di sfondo, ma dice a Vim
di che tipo è il colore di sfondo. Per cambiare il colore di sfondo,
vedere |**:hi-normal**|.

Quando 'background' è impostato Vim adatterà il gruppo di colori di
default per il nuovo valore. Ma i colori usati per l'evidenziazione
della sintassi non cambieranno. *g:colors_name*
Quando viene caricato uno schema di colori (viene impostata la
variabile "g:colors_name") impostare 'background' causerà un nuovo
caricamento dello schema dei colori. Se lo schema di colore viene
posto al valore di 'background' ciò funzionerà.
Comunque, se lo schema di colore imposta ancora 'background' l'effetto
può venire annullato. Prima cancellate la variabile "g:colors_name"
se necessario.

Impostando 'background' al valore di default con: >
:set background&
< Vim determinerà il valore. Nella GUI ciò dovrebbe funzionare bene,
in altri casi Vim potrebbe non riuscire a vedere il valore giusto.

Quando l'opzione |**t_RB**| è impostata, Vim la userà per ottenere
il valore del background nel terminale. Se il valore RGB che viene
restituito dal terminale è di tipo chiaro/scuro e il valore di
'background' non è chiaro/scuro, 'background' verrà impostato, e la
videata rinfrescata. Ciò può avere effetti collaterali: se si
presume di avere questo problema, impostare t_BG alla stringa nulla
nel file .vimrc. Il risultato può essere trovato nella variabile
|**v:termrbgresp**|.

Avviando la GUI, il valore di default per 'background' sarà "light".
Quando il valore non è impostato in .gvimrc, e Vim scopre che lo
sfondo è proprio scuro, 'background' viene impostato a "dark". Ma ciò
succede soltanto DOPO che sia stato letto il file .gvimrc (perché la
finestra deve venire aperta per scoprire il colore corrente dello
sfondo). Per superare ciò, forzate la finestra ad essere aperta
mettendo il comando ":gui" nel file .gvimrc, prima che venga usato il
valore di 'background' (ad esempio, prima di ":syntax on").

Per MS-DOS, Windows e OS/2 il valore di default è "dark".
Per altri sistemi operativi, "dark" è usato quando 'term' è impostato
a "linux", "screen.linux", "cygwin" o "putty", oppure se $COLORFGBG
indica un colore di sfondo scuro. Altrimenti il default è "light".

Il comando |**:terminal**| e la funzione |**term_start()**| usano il valore
di 'background' per decidere se la finestra di terminale sarà
inizializzata con uno sfondo bianco oppure nero.

Normalmente quest'opzione potrebbe essere impostata nel file .vimrc.
Possibilmente in relazione con il nome del terminale. Esempio: >
:if &term == "pcterm"
: set background=dark
:endif
< Quando quest'opzione è impostata, le impostazioni di default per i
gruppi di evidenziazione cambieranno. Per usare altre impostazioni,
mettete i comandi ":highlight" DOPO avere impostato l'opzione
'background'.
Quest'opzione è anche usata nel file "$VIMRUNTIME/syntax/syntax.vim"
per selezionare i colori per l'evidenziazione della sintassi. Dopo
aver cambiato quest'opzione, dovete caricare di nuovo syntax.vim
per ottenere il risultato. Ciò può essere fatto con ":syntax on".

'backspace' *'bs'*
'backspace' 'bs' string (default: "", impostato a "indent,eol,start"
in |**defaults.vim**|)
globale
{non in Vi}
Influenza il funzionamento di <BS>, <Del>, CTRL-W e CTRL-U in modo
Insert. È un elenco di parole separate da virgola. Ogni parola

regola il comportamento del tasto di backspace in questi casi:
valore effetto ~
indent consente di effettuare un backspace sull'autoindentatura
eol   consente di effettuare un backspace sulle interruzioni di
        riga (unisce le righe)
start   consente di effettuare un backspace prima del punto in cui
        era cominciato l'Inserimento.
        CTRL-W e CTRL-U si fermano all'inizio dell'inserimento.

Quando il valore non è impostato, viene usato il backspace
compatibilmente con Vi.

Per compatibilità all'indietro con la versione 5.4 e anteriori:
valore effetto ~
  0   come ":set backspace=" (compatibile Vi)
  1   come ":set backspace=indent,eol"
  2   come ":set backspace=indent,eol,start"

Vedere |**:fixdel**| se i vostri tasti <BS> o <Del> non si comportano
come vi aspettate.
NOTA: Quest'opzione è impostata a "" se si attiva 'compatible'.

                              'backup' *'bk'* *'nobackup'* *'nobk'*
'backup' 'bk'   booleana   (default: off)
                        globale
                        {non in Vi}
Fare un backup prima di sovrascrivere un file. Tenerlo anche dopo che
il file è stato riscritto. Se non volete tenere il file di backup, ma
volete un backup mentre il file viene scritto, mettete a off questa
opzione, ed impostate l'opzione 'writebackup' (è questo il
comportamento di default). Se proprio non volete un file di backup,
azzerate entrambe le opzioni (utile se il vostro File System è quasi
pieno). Vedere |**backup-table**| per ulteriori spiegazioni.
Quando il file corrisponde alle specifiche di 'backupskip', un backup
non viene eseguito in nessun caso.
Quando l'opzione 'patchmode' è attiva, il backup può essere rinominato
e divenire la versione precedente di un file.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

                                              'backupcopy' *'bkc'*
'backupcopy' 'bkc'   stringa (default di Vi in Unix: "yes", se no: "auto")
                        globale o locale a un buffer |**global-local**|
                        {non in Vi}
Quando si scrive un file e viene richiesto un backup, quest'opzione
dice come va fatto. Questa è una lista di parole, separate da
virgola.

I valori principali sono:
"yes"   fare una copia del file e riscrivere il file originale
"no"   rinominare il file e scriverne uno nuovo
"auto" una delle due opzioni precedenti, la più efficiente

Valori ulteriori, che possono essere aggiunti a quelli elencati
sopra sono:
"breaksymlink" annullare sempre i link simbolici in scrittura
"breakhardlink" annullare sempre i link "hard" in scrittura

Fare una copia del file e riscrivere sul file originale:
- Richiede più tempo per effettuare la copia del file.
+ Quando il file ha attributi speciali, è un link (hard/simbolico) o
  ha una "resource fork" [Macintosh], li mantiene.
- Quando il file è un link, il backup avrà il nome del link, non del
  file vero.

Rinominare il file e scriverne uno nuovo:
+ È veloce.
- Talora non tutti gli attributi del file originale possono essere
  copiati nel nuovo file.
- Quando il file originale è un link, il nuovo file NON sarà un link.

Il valore "auto" è la via di mezzo: Quando Vim vede che un "rename"
del file privo di effetti indesiderati è possibile (gli attributi
possono essere riprodotti e il file non è un link) lo utilizza.

Quando ci si aspetta che questo possa causare dei problemi, verrà effettuata una copia.

I valori "breaksymlink" e "breakhardlink" si possono usare in qualsiasi combinazione uniti a "yes", o a "no", o a "auto". Se specificati richiedono a Vim di annullare i link sia simbolici che "hard"facendo esattamente quel che fa l'opzione "no", ossia rinominando il file originale perché faccia da backup, e scrivendo un file nuovo al suo posto. Ciò può essere utile per esempio in strutture ad albero composte da file di sorgenti, in cui ogni file è un link simbolico o "hard" e ogni modifica deve rimanere soltanto nell'albero di file sorgenti locale, senza venire propagata all'indietro all'albero sorgente originale.

crontab
Una situazione in cui "no" e "auto" daranno problemi: Un programma che apra un file, richiami Vim per editare il file e poi controlli se il file aperto è stato cambiato (utilizzando la descrizione del file) controllerà il file di backup, invece del file nuovo creato da Vim. "crontab -e" è un esempio.

Dopo aver effettuata una copia, il file originale viene svuotato e quindi riempito con il nuovo testo [risultante dalle modifiche apportate con Vim - NdT]. Questo implica che la maschera di protezione, il proprietario e i link simbolici del file originale sono gli stessi di prima. Il file di backup, invece, è un nuovo file, il cui proprietario è l'utente che sta editando il file. Il gruppo di appartenenza del file di backup è lo stesso gruppo del file originale. Se l'assegnazione [al gruppo] non è consentita, gli accessi consentiti al gruppo sono gli stessi consentiti al "resto del mondo".

Quando il file viene rinominato, la situazione è quella opposta: Il file di backup ha gli stessi attributi del file originale, e il file scritto da Vim ha come proprietario l'utente che sta facendo la modifica. Se il file originale era un link (hard/simbolico), il nuovo file non lo è! Per questa ragione l'impostazione "auto" non rinomina il file quando si tratta di un link. Il proprietario e il gruppo del file scritto da Vim saranno gli stessi del file originale, ma il sistema operativo potrebbe non consentirlo. Nel qual caso l'impostazione "auto" si asterrà dal rinominare il file.

NOTA: Quest'opzione è impostata al valore di default di Vi quando si attiva 'compatible' e al valore di default di Vim quando si imposta a off 'compatible'.

'backupdir' *'bdir'*
'backupdir' 'bdir' stringa (default per Amiga: ".,t:",
per MS-DOS e Win32: ".,$TEMP,c:/tmp,c:/temp"
per Unix: ".,~/tmp,~/")
globale
{non in Vi}
Lista di directory per il file di backup, separate da virgole.
- Il file di backup sarà creato nella prima directory della lista in cui è possibile farlo. La directory deve essere già esistente, Vim non la creerà per voi.
- Una lista vuota previene la creazione di un file di backup ('patchmode' non è possibile!). La scrittura del file può andare male per questo motivo.
- Specificando come directory "." si chiede di mettere il file di backup nella stessa directory del file che si sta editando.
- Una directory il cui nome inizia con "./" (o ".\" per MS-DOS et al.) chiede di mettere il file di backup in una directory a partire da quella in cui si trova il file editato. Il "." iniziale è rimpiazzato con il nome della directory in cui si trova il file editato.
("." dentro un nome di directory non ha un significato particolare).
- Spazi dopo la virgola sono ignorati, gli altri spazi sono considerati parte del nome della directory. Per inserire uno spazio come primo carattere in un nome di directory, prefissatelo con '\'.
- Per includere una virgola in un nome di directory, prefissatelo con un '\'.
- Un nome di directory può finire con '/'.
- Le variabili d'ambiente vengono valutate |**:set_env**|.

```
        - Attenzione ai caratteri '\', inserirne uno prima di uno spazio,
          inserirne due per ottenerne uno solo come valore nell'opzione
          (vedere |option-backslash|), ad esempio: >
            :set bdir=c:\\tmp,\ dir\\,with\\,commas,\\\ dir\ with\ spaces
<       - Per compatibilità all'indietro con Vim version 3.0 un '>' a inizio
          opzione è rimosso.
        Vedere anche le opzioni 'backup' e 'writebackup'.
        Se volete che i vostri file di backup in Unix restino nascosti, potete
        specificare: >
                :set backupdir=./.backup,~/.backup,.,/tmp
<       Va creata una directory ".backup" in ogni directory e nella vostra
        home directory perché una simile impostazione funzioni correttamente.
        L'uso di |:set+=| e |:set-=| è da preferire per aggiungere o togliere
        directory dalla lista.  Questo evita problemi nel caso una futura
        versione abbia un altro valore di default.
        Quest'opzione non può essere impostata da |modeline|, o nel |sandbox|,
        per motivi di sicurezza.

                                                *'backupext'* *'bex'* *E589*
'backupext' 'bex'       stringa (default: "~", per VMS: "_")
                        globale
                        {non in Vi}
        Stringa da aggiungere a un nome di file per farne il nome del file di
        backup.  Il default è piuttosto insolito, per evitare ricoprire per
        errore con un file di backup dei file esistenti.  Si può scegliere di
        usare ".bak", ma assicuratevi di non avere file con suffisso ".bak"
        che vi interessi tenere.
        Solo caratteri normali possono essere usati nei nomi di file, i
        caratteri "/\*?[|<>" non sono validi.

        Se preferite tenere tanti backup, potreste usare un autocomando per
        l'evento BufWritePre per cambiare 'backupext', immediatamente prima
        della scrittura del file, per includere un'indicazione della data. >
                :au BufWritePre * let &bex = '-' . strftime("%Y%b%d%X") . '~'
<       Usare 'backupdir' per posizionare il file di backup in una directory
        differente.
                                                *'backupskip'* *'bsk'*
'backupskip' 'bsk'      stringa (default: "$TMPDIR/*,$TMP/*,$TEMP/*"
                                 Unix: "/tmp/*,$TMPDIR/*,$TMP/*,$TEMP/*"
                                 Mac: "/private/tmp/*,$TMPDIR/*,$TMP/*,$TEMP/*")
                        globale
                        {non in Vi}
                        {non disponibile se compilato senza la funzionalità
                        |+wildignore|}
        Una lista di modelli di nomi di file.  Quando uno dei modelli
        corrisponde al nome di un file che si vuole scrivere, non viene creato
        alcun file di backup.  Sia il nome del file specificato che il suo
        nome completato con il nome della directory in cui si trova sono
        utilizzati.  Il modello di nome è usato come con |:autocmd|, vedere
        |autocmd-patterns|.
        Attenti ai caratteri speciali, vedere |option-backslash|.
        Quando $TMPDIR, $TMP o $TEMP non sono definiti, non vengono usati come
        valori di default.  "/tmp/*" è usato solo sotto Unix.

        ATTENZIONE: L'assenza di un file di backup implica che se Vim non
        riesce a scrivere il vostro buffer correttamente, e in seguito, per
        qualsiasi ragione, Vim termina, voi perdete sia il file originale che
        quello che stavate scrivendo.  Inibite i backup solo se la perdita
        eventuale del file non è importante per voi.

        Nota: Le variabili d'ambiente non sono valutate.  Se volete usare
        $HOME dovete espanderla esplicitamente, ad es.: >
                :let &backupskip = escape(expand('$HOME'), '\') . '/tmp/*'

<       Nota: Il default assicura anche che il comando Unix "crontab -e"
        funzioni come deve (se si fosse fatto un backup rinominando il file
        originale, "crontab" non avrebbe visto il file di nuova creazione).
        Vedere anche 'backupcopy' e |crontab|.
<
                                                *'balloondelay'* *'bdlay'*
'balloondelay' 'bdlay'  numero  (default: 600)
                        globale
                        {non in Vi}
```

{disponibile solo se compilato con la funzionalità
|+balloon_eval|}
Ritardo in millisecondi prima di visualizzare una didascalia. Vedere
|balloon-eval|.

'ballooneval' *'beval'* *'noballooneval'* *'nobeval'*
'ballooneval' 'beval' booleana (default: off)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+balloon_eval|}
Attiva la funzionalità didascalia |balloon-eval| per la GUI.

'balloonevalterm' *'bevalterm'* *'noballoonevalterm'*
'nobevalterm'
'balloonevalterm' 'bevalterm' booleana (default: off)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+balloon_eval_term|}
Attiva la funzionalità didascalia |balloon-eval| per il terminale.

'balloonexpr' *'bexpr'*
'balloonexpr' 'bexpr' stringa (default: "")
globale o locale a un buffer |global-local|
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+balloon_eval|}
Espressione per il testo da mostrare in una didascalia di valutazione.
Usata solo se 'ballooneval' è attivata. Le seguenti variabili possono
essere usate:

v:beval_bufnr numero del buffer in cui la didascalia apparirà
v:beval_winnr numero della finestra
v:beval_winid ID della finestra
v:beval_lnum numero della riga
v:beval_col numero della colonna (indice del byte)
v:beval_text parola sotto o dopo il puntatore del mouse

La valutazione dell'espressione non deve avere effetti collaterali!
Esempio: >

```
function! MyBalloonExpr()
    return 'Il cursore è alla riga ' . v:beval_lnum .
            \', colonna ' . v:beval_col .
            \ ' del file ' .  bufname(v:beval_bufnr) .
            \ ' sulla parola "' . v:beval_text . '"'
endfunction
set bexpr=MyBalloonExpr()
set ballooneval
```

<

Vedere anche |balloon_show()|, che si può usare se il contenuto del
della didascalia va inserito in modo asincrono.

NOTA: La didascalia è visualizzata solo se il cursoe si trova su
un carattere di testo. Se il risultato della valutazione di
'balloonexpr' non è la stringa nulla, Vim non tenta di spedire un
messaggio a un debugger esterno (Netbeans o Sun Workshop).

L'espressione sarà valutata nel |sandbox|, se è stata impostata da una
modeline; vedere |sandbox-option|.

Non è consentito cambiare testo o passare a un'altra finestra mentre è
in corso la valutazione di 'balloonexpr' |textlock|.

Per controllare se ci sono più righe nel testo della didascalia, usare
questo controllo: >

```
        if has("balloon_multiline")
```

<
Se sono supportati, i caratteri "\n" andranno a capo. Se
l'espressione contiene il nome di una lista (List), ciò equivale a
usare ogni elemento della "List" come una stringa, mettendo "\n" dopo
ognuno di essi.
NOTA: Quest'opzione è impostata a "" se si attiva 'compatible'.

'belloff' *'bo'*

'belloff' 'bo' stringa (default "")
globale
{non in Vi}

Specifica per quali eventi non va suonato un campanello di avviso. È una lista di elementi separati da virgole. Per ogni elemento presente, il campanello non suonerà. Ciò è molto utile per specificare eventi da non segnalare mentre si sta operando in modo Insert.

elemento	significato quando compare nella lista ~
all	Tutti gli eventi.
backspace	Quando si immette <BS> o <Del> e la cancellazione produce un errore.
cursor	Movimento non riuscito usando i tasti del cursore o <PageUp>/<PageDown> in |**Insert-mode**|.
complete	Errore in caso di uso di |**i_CTRL-X_CTRL-K**| o |**i_CTRL-X_CTRL-T**|.
copy	Impossibile copiare caratteri in modo Insert usando|**i_CTRL-Y**| o |**i_CTRL-E**|.
ctrlg	Carattere sconosciuto dopo <C-G> in Insert mode.
error	Altra condizione di errore (p.es. se si chiede di unire la riga seguente, essendo posizionati sull'ultima linea) (usato principalmente in |**Normal-mode**| o |**Cmdline-mode**|).
esc	Immettere <Esc> in |**Normal-mode**|.
ex	In |**Visual-mode**|, immettere |**Q**| produce un errore.
hangul	Errore durante l'input in hangul (coreano).
insertmode	Immettere <Esc> in 'insertmode'.
lang	Richiamare il modulo "campanello" per Lua/Mzscheme/TCL.
mess	Nessun output disponibile per |**g<**|.
showmatch	Errore eseguendo la funzione 'showmatch'.
operator	Errore di regione vuota per |**cpo-E**|.
register	Registro sconosciuti dopo <C-R> in |**Insert-mode**|.
shell	Campanello in output dalla shell |**:!**|.
spell	Errore durante un suggerimento ortografico.
wildmode	Ulteriori corrispondenze disponibili in |**cmdline-completion**|(dipende dalle impostazioni di 'wildmode').

Quest'opzione è molto utile per decidere in quali particolari occasioni si dovrebbe suonare il campanello in modo Insert. Per i comandi nei modi Normal ed Ex, il campanello è suonato spesso per segnalare un un errore. Si può evitare che suoni aggiungendo l'elemento "error".

'binary' *'bin'* *'nobinary'* *'nobin'*

'binary' 'bin' booleana (default: off)
locale al buffer
{non in Vi}

Quest'opzione dovrebbe essere impostata prima di editare un file binario. Potete anche usare l'argomenti di Vim |**-b**|. Quando questa opzione è attivata, alcune altre opzioni saranno modificate (e sono attive anche quando si parte con l'opzione già attivata):

'textwidth' sarà impostata a 0
'wrapmargin' sarà impostato a 0
'modeline' sarà off
'expandtab' sarà off

Inoltre, le opzioni 'fileformat' e 'fileformats' non saranno usate, il file è letto e scritto come se 'fileformat' sia "unix" (un solo <NL> separa le righe).
Le opzioni 'fileencoding' e 'fileencodings' non saranno usate, il file è letto senza conversioni.
NOTA: Se si inizia ad editare un altro file mentre l'opzione 'bin' è attiva, impostazioni dagli autocomandi possono cambiare di nuovo le impostazioni (ad es., 'textwidth'), causando problemi in seguito.
Potreste impostare 'bin' nuovamente quando il file è stato caricato.
I precedenti valori di queste opzioni sono salvati e ripristinati quando l'opzione 'bin' è inattiva. Ogni buffer ha il suo proprio assieme di valori salvati delle opzioni.
Per editare un file avendo impostato 'binary', si può usare l'argomento |**++bin**|. Ciò evita di dover dare il comando ":set bin", il quale avrebbe effetto che tutti i file in modifica nella stessa sessione di Vim.

Nello scrivere un file il carattere <EOL> per l'ultima riga è scritto solo se era già presente nel file originale (normalmente Vim aggiunge un carattere <EOL> all'ultima riga, se questo manca; questo renderebbe il file più lungo). Vedere l'opzione 'endofline'.

'biosk' *'nobiosk'*
'biosk' 'nobiosk' booleana (default: on)
globale
{non in Vi} {solo per MS-DOS}
Quest'opzione riguardava MS-DOS e non è più supportata.

'bomb' *'nobomb'*
'bomb' booleana (default: off)
locale al buffer
{non in Vi}
{disponibile solo se compilato con la funzionalità |+multi_byte|}
Quando si scrive un file e sono soddisfatte le condizioni seguenti, un BOM (Byte Order Mark) è aggiunto all'inizio del file:
- quest'opzione è a on
- l'opzione 'binary' è a off
- 'fileencoding' è "utf-8", "ucs-2", "ucs-4" o una delle varianti little/big endian.

Alcune applicazioni usano il BOM per determinare la codifica del file. Usato spesso per file UCS-2 in MS-Windows. Per altre applicazioni può essere causa di problemi, ad esempio: "cat file1 file2" mette il BOM di file2 nel mezzo del file risultante. Gcc non accetta un BOM. Quando Vim legge un file e 'fileencodings' comincia con "ucs-bom", viene effettuato un controllo per la presenza del BOM e 'bomb' impostato di conseguenza.
Se 'binary' non è impostato, il BOM è rimosso dalla prima riga, così che non compaia quando si edita il file. Se non si cambiano le opzioni, il BOM viene riscritto ancora quando si scrive il file.

'breakat' *'brk'*
'breakat' 'brk' stringa (default: " ^I!@*-+;:,./?")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |+linebreak|}
Quest'opzione permette di scegliere quali caratteri possono causare un'interruzione di riga quando 'linebreak' è attivo. Funziona solo per i caratteri ASCII e anche per i caratteri a 8-bit, quando 'encoding' è a 8-bit.

'breakindent' *'bri'* *'nobreakindent'* *'nobri'*
'breakindent' 'bri' booleana (default: off)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità |+linebreak|}
Ogni linea ripiegata continuerà visivamente indentata (stessa quantità di spazio come la parte iniziale della linea stessa), preservando in questo modo i blocchi di testo orizzontali.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'breakindentopt' *'briopt'*
'breakindentopt' 'briopt' stringa (default: "")
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità |+linebreak|}
Impostazioni per 'breakindent'. Può consistere dei seguenti elementi opzionali, che devono essere separati da virgola.

min:{n}	Larghezza minima del testo che verrà conservato impostando 'breakindent', anche se il testo risultante sarebbe normalmente più stretto. Ciò fa sì che il testo indentato quasi al bordo destro della finestra non occupi molto spazio verticale quando la linea viene spezzata.
shift:{n}	Dopo aver impostato 'breakindent', l'inizio di ogni linea ripiegata sarà spostato di quel numero di caratteri. Ciò permette che l'indentatura di paragrafo dinamica di tipo

francese (se negativo) o di rimarcare la
continuazione della linea (se positivo).
sbr Visualizza il valore di 'showbreak' prima di
applicare l'ulteriore indentatura.
Il valore di default per min è 20 e per shift è 0.

'browsedir' *'bsdir'*
'browsedir' 'bsdir' stringa (default: "last")
globale
{non in Vi} {solo per le GUI Motif, Athena, GTK, Mac e
Win32}
Che directory usare per il file browser:
last Usare la stessa directory dell'ultimo file browser, in
cui un file sia stato salvato.
buffer Usare la directory del buffer corrispondente.
current Usare la directory corrente.
{path} Usare la directory specificata.

'bufhidden' *'bh'*
'bufhidden' 'bh' stringa (default: "")
locale al buffer
{non in Vi}
Quest'opzione specifica cosa succede quando un buffer non è più
visualizzato in una finestra:
<empty> usare il valore dell'opzione globale 'hidden'
hide nascondere il buffer (senza scaricarlo) anche quando
'hidden' non è impostato
unload scaricare il buffer, anche quando 'hidden' è impostato
o si sta usando |**:hide**|
delete toglie il buffer dalla lista dei buffer, anche quando
'hidden' è impostato o si sta usando |**:hide**|,
equivale a usare |**:bdelete**|
wipe cancella il buffer dalla lista dei buffer, anche quando
'hidden' è impostato o si sta usando |**:hide**|,
equivale a usare |**:bwipeout**|

ATTENZIONE: quando si usa "unload", "delete" o "wipe", le modifiche a
un buffer vanno perse senza alcun messaggio di avvertimento. Inoltre
questi valori possono impedire il funzionamento corretto di
autocomandi che passano temporaneamente da un buffer all'altro.
Quest'opzione è usata insieme a 'buftype' e 'swapfile' per
specificare tipi speciali di buffer. Vedere |**special-buffers**|.

'buflisted' *'bl'* *'nobuflisted'* *'nobl'* *E85*
'buflisted' 'bl' booleana (default: on)
locale al buffer
{non in Vi}
Quando quest'opzione è impostata, il buffer è visibile nella lista
dei buffer. Se vale off non è usata per ":bnext", "ls", il menù
Buffers, etc.
Quest'opzione è messa a off da Vim per buffer che sono usati solo per
ricordare un nome di file o dei mark. Vim la imposta quando si inizia
a editare un buffer. Non viene alterata quando si passa ad un nuovo
buffer con ":buffer".

'buftype' *'bt'* *E382*
'buftype' 'bt' stringa (default: "")
locale al buffer
{non in Vi}
Il valore di quest'opzione specifica di che tipo è un buffer:
<empty> buffer normale
nofile buffer non collegato a un file, e da non scrivere
nowrite buffer da non scrivere
acwrite buffer che sarà sempre scritto con autocomandi
innescati da BufWriteCmd.
quickfix buffer tipo quickfix, contiene una lista di errori
|**:cwindow**| o una lista di posizioni |**:lwindow**|
help buffer di aiuto (di solito non impostato manualmente)
terminal buffer per un terminale |**terminal**| (di solito non
impostato manualmente)

Quest'opzione è usata insieme a 'bufhidden' e 'swapfile' per
specificare tipi speciali di buffer. Vedere |**special-buffers**|.

Attenzione a cambiare quest'opzione, può avere effetti collaterali!

Un buffer di tipo "quickfix" è usato solo per una lista di errori e una lista di posizioni.
Questo valore sono impostati dai comandi |**:cwindow**| e |**lwindow**| e non ci si aspetta che vengano cambiati dall'utente.

I buffer di tipo "nofile" e "nowrite" sono simili:
entrambi: Il buffer non è scritto a disco, ":w" non funziona (":w nome_file" invece funziona).
entrambi: Il buffer non si considera mai modificato |**'modified'**|. Non ci sono preavvisi quando si perdono modifiche, ad esempio se uscite da Vim.
entrambi: Un file di swap è creato solo se si sta usando troppa memoria (se 'swapfile' è stato messo off non si usa mai un file di swap).
solo nofile: Il nome del buffer è fisso, non è gestito come un nome di file. Non si modifica a fronte di un comando |**:cd**|.
entrambi: Se si usa ":e nome-buffer" e si sta già editando "nome-buffer", il buffer è svuotato, e sono innescati gli autocomandi, come si fa solitamente per |**:edit**|.

E676

"acwrite" implica che il nome del buffer non sia collegato a un nome di file, come è anche il caso per "nofile", ma che sarà riscritto. Quindi, a differenza del caso di "nofile" e "nowrite", ":w" viene eseguito, e un buffer modificato non può essere scartato senza che sia stato salvato. Per scriverlo, dev'esserci un autocomando |**BufWriteCmd**|, |**FileWriteCmd**| o |**FileAppendCmd**|.

'casemap' *'cmp'*

'casemap' 'cmp' stringa (default: "internal,keepascii")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+multi_byte**|}

Specifica dettagli riguardo al cambiamento maiuscolo/minuscolo.
Può contenere queste parole, separate da virgola:
internal Usa le funzioni interne di mappatura, la lingua locale corrente non cambia la mappatura minuscolo/maiuscolo. Questo ha importanza solamente quando 'encoding' è una codifica Unicode, "latin1" o "iso-8859-15". Quando "internal" è omesso, le funzioni di sistema towupper() e towlower() vengono usate, se disponibili.
keepascii Per i caratteri ASCII (da 0x00 a 0x7f) usa la mappatura US, il locale corrente non ha effetto. Ha importanza probabilmente solo per il Turco.

'cdpath' *'cd'* *E344* *E346*

'cdpath' 'cd' stringa (default: equivalente a $CDPATH o ",,")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+file_in_path**|}

Questa è una lista di directory che verranno cercate quando si usano i comandi |**:cd**| e |**:lcd**|, a condizione che la directory cercata abbia un nome in formato relativo, non un nome assoluto, che inizia con "/", "./" o "../", nel qual caso l'opzione 'cdpath' non è usata.
Il valore dell'opzione 'cdpath' ha la stessa forma e semantica di |**'path'**|. Vedere anche |**file-searching**|.
Il valore di default è preso da $CDPATH, con una "," preposta per cercare prima nella directory corrente.
Se il valore di default preso da $CDPATH non è quello desiderato, si includa una versione modificata del seguente comando nel file vimrc per reimpostarlo: >

```
:let &cdpath = ',' . substitute(substitute($CDPATH, '[, ]', '\\\0', 'g'), ':', ',', 'g')
```

< Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza (parti di 'cdpath' possono essere passate alla shell per espandere i nomi dei file).

'cedit'
'cedit' stringa (default Vi: "", default Vim: CTRL-F)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|+vertsplit|}
Il tasto usato in modo Command line per aprire la finestra riga
di comando. Il default è CTRL-F quando 'compatible' non è attivo.
Si possono scegliere solo tasti non-stampabili.
Il tasto può essere specificato come singolo carattere, ma è difficile
da digitare. Si consiglia di usare la notazione <> . Ad es.: >
:exe "set cedit=\<C-Y>"
:exe "set cedit=\<Esc>"
< |Nvi| ha quest'opzione, ma usa solo il primo carattere.
vedere |cmdwin|.
NOTA: Quest'opzione è impostata al valore di default di Vim quando si
imposta a off 'compatible'.

'charconvert' *'ccv'* *E202* *E214* *E513*
'charconvert' 'ccv' stringa (default: "")
globale
{disponibile solo se compilato con le funzionalità
|+multi_byte| ed |+eval|}
{non in Vi}
Un'espressione usata per convertire la codifica dei caratteri. Viene
valutata quando un file che deve essere letto o è stato scritto ha una
codifica diversa da quella desiderata.
'charconvert' non è usato quando la funzione interna iconv() è
supportata e può fare la conversione. L'uso di iconv() è preferibile,
in quanto molto più veloce.
'charconvert' non è usato durante la lettura di "stdin" |--|, perché
manca il file da convertire. Si dovrà salvare il testo in un file in
precedenza.
L'espressione restituirà zero o una stringa nulla in caso di riuscita,
restituirà non-zero in caso di insuccesso.
I possibili nomi di codifiche incontrati sono in 'encoding'.
Inoltre, sono usati i nomi dati in 'fileencodings' e 'fileencoding'.
Le conversioni tra "latin1", "unicode", "ucs-2", "ucs-4" e "utf-8"
sono fatte internamente da Vim, 'charconvert' non viene usato.
'charconvert' è anche usato per convertire il file viminfo, se
l'opzione 'c' è presente in 'viminfo'. Usato anche per conversione
Unicode.
Esempio: >
set charconvert=CharConvert()
fun CharConvert()
system("recode "
\ . v:charconvert_from . ".." . v:charconvert_to
\ . " <" . v:fname_in . " >" v:fname_out)
return v:shell_error
endfun
< Le relative variabili di Vim sono:
v:charconvert_from nome della codifica corrente
v:charconvert_to nome della codifica desiderata
v:fname_in nome del file di input
v:fname_out nome del file di output
Nota v:fname_in e v:fname_out non saranno mai uguali.
Nota v:charconvert_from e v:charconvert_to possono essere diversi
da 'encoding'. Vim utilizza internamente UTF-8 invece che UCS-2 o
UCS-4. Vim non esegue cifratura quando si usa 'charconvert'. Se si
vuole criptare il file dopo la conversione, questo dovrebbe essere
fatto in 'charconvert'.
Quest'opzione non può essere impostata da |modeline|, o nel |sandbox|,
per motivi di sicurezza.

'cindent' *'cin'* *'nocindent'* *'nocin'*
'cindent' 'cin' booleana (default: off)
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità
|+cindent|}
Abilita l'indentatura automatica di programmi C. Vedere 'cinkeys' per
impostare i tasti che applicano l'indentatura in modo Insert e
'cinoptions' per impostare lo stile di indentatura preferito.

Se 'indentexpr' non è nullo, prevale su 'cindent'.
Se 'lisp' non è a on e sia 'indentexpr' che 'equalprg' sono nulli,
l'operatore "=" applica l'indentatura usando questo algoritmo
invece di chiamare un programma esterno.
Vedere |**C-indenting**|.
Se non si gradisce il modo in cui funziona 'cindent', si provi
l'opzione 'smartindent' o 'indentexpr'.
Quest'opzione non viene usata quando 'paste' è impostato.
NOTA: Quest'opzione viene reimpostata quando 'compatible' è messo
a on.

'cinkeys' *'cink'*
'cinkeys' 'cink' stringa (default: "0{,0},0),:,0#,!^F,o,O,e")
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+cindent**|}
Una lista di tasti che, quando digitati in modo Insert causano
l'indentatura della riga corrente. Si usa solo se 'cindent' è attivo
e 'indentexpr' è nullo.
Per il formato di questa opzioni vedere |**cinkeys-format**|.
Vedere |**C-indenting**|.

'cinoptions' *'cino'*
'cinoptions' 'cino' stringa (default: "")
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+cindent**|}
'cinoptions' stabilisce il modo in cui 'cindent' rientra le righe in
un programma C. Vedere |**cinoptions-values**| per i valori di questa
opzione, e |**C-indenting**| per informazioni sulle rientranze in C.

'cinwords' *'cinw'*
'cinwords' 'cinw' stringa (default: "if,else,while,do,for,switch")
locale al buffer
{non in Vi}
{non disponibile se compilato senza le due
funzionalità |**+cindent**| e |**+smartindent**|}
Le parole in questa lista aggiungono un'indentatura nella riga
seguente quando 'smartindent' o 'cindent' sono impostati. Per
'cindent' questo viene effettuato solo in una posizione appropriata
(all'interno di due {}).
Si noti che 'ignorecase' non è usato per 'cinwords'. Se va trascurato
il maiuscolo/minuscolo, si includano i comandi sia in maiuscolo che in
minuscolo: "if,If,IF".

'clipboard' *'cb'*
'clipboard' 'cb' stringa (default: "autoselect,exclude:cons\|Linux"
per X-windows, "" per il
resto)
globale
{non in Vi}
{solo nelle versioni GUI o quando
è inclusa la funzionalità |**+xterm_clipboard**|}
Quest'opzione è una lista di nomi separati da virgola.
Questi nomi sono riconosciuti:

clipboard-unnamed
unnamed Quando incluso, Vim userà il registro degli appunti
'*' per tutte le operazioni (copia, elimina, cambia,
inserisci) che normalmente andrebbero nel registro
senza nome. Quando un registro è specificato, sarà
usato sempre sia quando "unnamed" è in 'clipboard'
sia quando non lo è.
Il registro degli appunti può venire consultato
usando la notazione "*. Vedere anche |**gui-clipboard**|.

clipboard-unnamedplus
unnamedplus Una variante del flag "unnamed" che usa il registro
'clipboard' '+' (|**quoteplus**|) invece che il register
'*' per tutte le operazioni, di copia, cancellazione,

modifica e incolla, che normalmente finirebbero nel registro "unnamed". Quando "unnamed" è pure specificato nell'opzione, le operazioni di yank [copia] (ma non quelle di cancellazione, modifica o incolla) copieranno il testo anche nel registro '*'. Disponibile solo con la funzionalità |**+X11**|. La disponibilità può essere controllata con: >

```
        if has('unnamedplus')
<
```

clipboard-autoselect

autoselect — Funziona come l'opzione 'a' in 'guioptions': Se presente, ogni volta che si entra in modo Visual, o l'area Visuale viene estesa, Vim cerca di diventare il proprietario della selezione globale del sistema a finestre o di mettere il testo selezionato negli appunti usati dal registro di selezione "*. Vedere |**guioptions_a**| e |**quotestar**| per ulteriori dettagli. Quando la GUI è attiva, è usata l'opzione 'a' in 'guioptions', quando la GUI non è attiva, questa opzione "autoselect" viene usata. Si applica anche alla selezione senza modalità.

clipboard-autoselectplus

autoselectplus — Come "autoselect" ma utilizzando il registro "+ invece che il registro "*. Confrontare con il flag 'P' in 'guioptions'.

clipboard-autoselectml

autoselectml — Come "autoselect", ma vale solo per la selezione senza modalità. Paragonabile all'opzione 'A' in 'guioptions'.

clipboard-html

html — Quando la 'clipboard' contiene del codice HTML, usare questo quando si effettua "incolla". Quando si mette testo nell 'clipboard', viene marcato come HTML. Questo può servire per copiare HTML visualizzato da Firefox, incollarlo come HTML grezzo in Vim, selezionare HTML in Vim e incollarlo in una finestra di "rich edit" in Firefox. È probabile che questo tipo di aggiunta sia solo provvisoria, se possibile usate autocomandi BufEnter. Solo supportata per GTK versione 2 e successive. disponibile solo con la funzionalità |**+multi_byte**|.

clipboard-exclude

exclude:{pattern} — Definisce un modello che viene confrontato col nome del terminale 'term'. Se il confronto ha successo, non verrà effettuata la connessione con il server X. Ciò è utile in questa situazione:
- Richiamando Vim da console.
- $DISPLAY è impostata in modo da lanciare applicazioni in un'altra finestra.
- Non si vuole effettuare la connessione al server X in console, ma in un emulatore di un terminale.

Per non connettersi mai al server X, usare: >

```
        exclude:.*
<
```

Ha lo stesso effetto dell'argomento |**-X**|. Nota Quando non c'è connessione con il server X il titolo della finestra non sarà ripristinato e gli appunti non saranno disponibili. Il valore di 'magic' è ignorato, {pattern} è interpretato come se 'magic' fosse attivo. Il resto del valore dell'opzione sarà considerato come {pattern}, che deve essere l'ultima voce.

'cmdheight' *'ch'*

'cmdheight' 'ch' — numero (default: 1) globale {non in Vi}

Numero delle righe da usare nella riga di comando. Aiuta ad evitare le richieste di input per proseguire |**hit-enter**|. Il valore di quest'opzione è memorizzato assieme alla pagina delle

linguette, in modo che ogni linguetta possa avere un valore differente.

'cmdwinheight' *'cwh'*
'cmdwinheight' 'cwh' numero (default: 7)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |+vertsplit|}

Numero delle righe da usare nella finestra contenente righe di comando. |cmdwin|

'colorcolumn' *'cc'*
'colorcolumn' 'cc' stringa (default "")
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità |+syntax|}

'colorcolumn' è una lista, separata da virgole, di colonne dello schermo che sono evidenziate con ColorColumn |hl-ColorColumn|. Utile per allineare del testo. Rallenterà la visualizzazione dello schermo. Il numero di colonna può essere un numero intero, oppure un numero preceduto dai segni '+' o '-', da aggiungere a (o sottrarre da) 'textwidth'. >

```
:set cc=+1  " evidenzia colonna dopo 'textwidth'
:set cc=+1,+2,+3  " evidenzia tre colonne dopo 'textwidth'
:hi ColorColumn ctermbg=lightgrey guibg=lightgrey
```
<

Se 'textwidth' vale zero i numeri preceduti da '-' e '+' sono ignorati.
Si possono evidenziare al massimo 256 colonne.

'columns' *'co'* *E594*
'columns' 'co' numero (default: 80 o larghezza terminale)
globale
{non in Vi}

Numero di colonne dello schermo. Normalmente viene impostato inizializzando il terminale e non c'è bisogno di impostarlo a mano. Vedere anche |posix-screen-size|.
Quando Vim sta lavorando con una GUI o in una finestra a dimensioni variabili, cambiando quest'opzione si causerà il ridimensionamento della finestra. Quando si vuole solo usare la grandezza per la GUI, inserire il comando nel file |gvimrc|.
Quando viene impostata quest'opzione e Vim non riesce a cambiare il numero di colonne dello schermo, lo schermo potrebbe dare risultati inaspettati. Usando la GUI questa modifica è sempre possibile e Vim limita il numero delle colonne a quelle che lo schermo può contenere. Potete usare questo comando per ottenere la finestra più grande possibile: >

```
:set columns=9999
```
<

Il valore minimo è 12, quello massimo 10000.

'comments' *'com'* *E524* *E525*
'comments' 'com' stringa (default
"s1:/*,mb:*,ex:*/,://,b:#,:%,:XCOMM,n:>,fb:-")
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità |+comments|}

Una lista di stringhe che possono iniziare una riga di commenti, separate da virgola. Vedere |format-comments|.
Vedere |option-backslash| per inserire spazi e backslash.

'commentstring' *'cms'* *E537*
'commentstring' 'cms' stringa (default: "/*%s*/")
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità |+folding|}

Un modello per il commento. La variabile "%s" è cambiata con il testo del commento. Attualmente è solo usato per aggiungere marcatori per la piegatura (folding), vedere |fold-marker|.

'compatible' *'cp'* *'nocompatible'* *'nocp'*
'compatible' 'cp' booleana (attiva per default, ma inattiva in presenza di un file |**vimrc**| o |**gvimrc**|, impostata a off in |**defaults.vim**|)
globale
{non in Vi}

Quest'opzione rende Vim più compatibile con Vi, oppure fa comportare Vim in un modo più utile.

Questa è un tipo di opzione speciale, perché quando è impostata o messa a off, altre opzioni vengono cambiate di conseguenza. ATTENZIONE: Impostare o mettere a off quest'opzione può causare molti effetti indesiderati: le mappature vengono interpretate in modo diverso, "annulla" [u] si comporta in modo diverso, ecc. Se si specifica quest'opzione nel file vimrc, è probabilmente meglio inserirla proprio all'inizio del file.

Per default quest'opzione è attivata e vengono usati i valori di default propri di Vi per le opzioni. È stato scelto questo valore di default per chi vuole usare Vim come userebbero Vi, e non è neppure interessato all'opzione 'compatible'.
Quando viene trovato un file |.**vimrc**| o |**gvimrc**| durante l'inizializzazione di Vim, quest'opzione viene messa a off, e tutte le opzioni non specificate saranno impostate coi default di Vim. In effetti, questo significa che quando un file |.**vimrc**| o |**gvimrc**| esiste, Vim userà i default di Vim, altrimenti userà i default di Vi. (Nota: Questo non accade per il file vimrc o gvimrc valido per l'intero sistema, e neppure per un file specificato con l'argomento |**-u**|). Vedere anche |**compatible-default**| e |**posix-compliance**|.
Si può anche impostare quest'opzione con l'argomento "-C" , ed metterla a off con "-N". Vedere |**-C**| e |**-N**|.
Vedere 'cpoptions' per ulteriori informazioni riguardo alla compatibilità con Vi.

Quando si imposta quest'opzione, numerose altre opzioni saranno impostate o messe a off per rendere Vim quanto più possibile compatibile con Vi. La tabella sottostante elenca tutte le opzioni interessate.
La colonna {?} indica quando una data opzione sarà modificata:
+ Significa che l'opzione è impostata al valore specificato nella colonna {impostato} quando 'compatible' è attivato.
& Significa che l'opzione è impostata al valore specificato in {impostato} quando 'compatible' è attivato, ed è impostata al suo valore di default Vim quando 'compatible' è messo a off.
- Significa che l'opzione NON è modificata quando si attiva 'compatible', ma che È impostata al suo valore di default Vim quando 'compatible' è messo a off.
La colonna {effetto} descrive la modifica posta in essere quando si attiva 'compatible'.

opzione	?	impostato	effetto ~
'allowrevins'	+	off	non c'è il comando CTRL-_
'antialias'	+	off	non usare caratteri antialias
'arabic'	+	off	annulla opzioni relative all'arabico
'arabicshape'	+	on	correggi forme dei caratteri
'backspace'	+	""	backspace normale
'backup'	+	off	nessun file di backup
'backupcopy'	&	Unix: "yes" resto: "auto"	il file di backup è una copia copia o rinomina il file di backup
'balloonexpr'	+	""	testo da mostrare nella didascalia
'breakindent'	+	off	niente indentatura per linee piegate
'cedit'	-	{come prima}	{imposta default vim se 'cp' è a off}
'cindent'	+	off	nessuna indentatura del codice C
'compatible'	-	{come prima}	{imposta default vim se 'cp' è a off}
'copyindent'	+	off	non copiare struttura indentatura
'cpoptions'	&	(ogni flag)	flag per compatibilità-Vi
'cscopepathcomp'	+	0	non mostrare directory in liste tag
'cscoperelative'	+	off	non usare "basename" del percorso come prefisso
'cscopetag'	+	off	non usare cscope per ":tag"

```
'cscopetagorder'+ 0             vedere |cscopetagorder|
'cscopeverbose' + off           vedere |cscopeverbose|
'delcombine'    + off           unicode: cancella carattere intero
'digraph'       + off           nessun digramma
'esckeys'       & off           nessun tasto-<Esc>in modo Insert
'expandtab'     + off           tab non estesi con spazi
'fileformats'   & ""            niente determinazione automatica
                                del formato del file,
                  "dos,unix"    tranne che per DOS, Windows e OS/2
'formatexpr'    + ""            usa 'formatprg' per auto-formattazione
'formatoptions' & "vt"          formattazione compatibile con Vi
'gdefault'      + off           opzione 'g' non di default per ":s"
'history'       & 0             nessuna cronologia della riga di
                                        comando
'hkmap'         + off           nessuna mappatura della tastiera
                                        per l'Ebreo
'hkmapp'        + off           nessuna mappatura della tastiera
                                        per l'Ebreo fonetico
'hlsearch'      + off           nessuna evidenziazione dei risultati
                                        di una ricerca
'incsearch'     + off           nessuna ricerca incrementale
'indentexpr'    + ""            nessuna indentatura per espressione
'insertmode'    + off           non iniziare in modo Insert
'iskeyword'     + "@,48-57,_"   le parole contengono caratteri
                                        alfanumerici e '_'
'joinspaces'    + on            inserire 2 spazi dopo il punto
'modeline'      & off           nessuna modeline
'more'          & off           nessuna pausa nelle liste
'mzquantum'     - {come prima}  {imposta default vim se 'cp' è a off}
'numberwidth'   & 8             numero minimo di colonne per
                                        numerazione righe
'preserveindent'+ off           non conservare la struttura di
                                        indentatura corrente
                                        in caso di modifiche
'revins'        + off           nessun inserimento inverso
'ruler'         + off           nessuna riga di informazioni sul file
'scrolljump'    + 1             nessun indicazione entità scorrimento
'scrolloff'     + 0             nessun bilanciamento righe attorno al
                                        cursore
'shelltemp'     - {come prima}  {imposta default vim se 'cp' è a off}
'shiftround'    + off           indentatura non arrotondata al valore
                                dell'opzione 'shiftwidth'
'shortmess'     & ""            nessuna abbreviazione dei messaggi
'showcmd'       & off           non mostra caratteri comandi digitati
'showmode'      & off           non mostra la modalità corrente
'sidescrolloff' + 0             il cursore va al bordo dello schermo
                                        durante lo scorrimento
'smartcase'     + off           non ignora automaticamente il cambio
                                        minuscolo/maiuscolo
'smartindent'   + off           nessuna indentatura automatica
'smarttab'      + off           nessun aggiustamento automatico
                                        tabulazione
'softtabstop'   + 0             tabulazione sempre posizionata usando
                                        'tabstop'
'startofline'   + on            va a inizio riga usando alcuni
                                        comandi
'tagcase'       & "followic"    'ignorecase' ricercando file di tag
'tagrelative'   & off           i nomi dei file nei tag non sono
                                        relativi
'termguicolors' + off           non evidenziare (guifg|guibg)
'textauto'      & off           non determinare automaticamente la
                                        modalità di testo
'textwidth'     + 0             nessuna andata a capo automatica
                                        per righe lunghe
'tildeop'       + off           la tilde [~] non è un operatore
'ttimeout'      + off           nessun timeout del terminale
'undofile'      + off           non usare un file undo (per
                                        ricordare annullamenti)
'viminfo'       - {come prima}  {imposta il valore di default di Vim solo
                                        se si imposta a off 'cp'}
'virtualedit'   + ""            il cursore può essere posizionato solo
                                        sui caratteri presenti
'whichwrap'     & ""            i movimenti a destra-sinistra non
```

fanno passare alla riga precedente/
seguente
'wildchar' & CTRL-E solo se il valore corrente è <Tab>
usa CTRL-E per il completamento dell
a
riga dei comandi
'writebackup' + on o off dipende dal valore della funzionalità
|+writebackup|

'complete' *'cpt'* *E535*
'complete' 'cpt' stringa (default: ".,w,b,u,t,i")
locale al buffer
{non in Vi}
Quest'opzione specifica come funziona il completamento dei comandi
|ins-completion| quando CTRL-P o CTRL-N sono usati. Viene anche usato
per il completamento di righe intere |i_CTRL-X_CTRL-L|. Indica il
tipo di completamento e i posti da controllare. È una lista di
opzioni separate da virgola:
. controlla il buffer corrente ('wrapscan' è ignorato)
w controlla i buffer da altre finestre
b controlla altri buffer caricati presenti nella lista buffer
u controlla i buffer non caricati presenti in lista buffer
U controlla i buffer non presenti nella lista dei buffer
k controlla i file dati con l'opzione 'dictionary'
kspell usa il controllo ortografico correntemente attivo |spell|
k{dict} controlla il file {dict}. Si possono specificare diverse
opzioni "k", anche parametrizzate. ad es.: >
:set cpt=k/usr/dict/*,k~/spanish
< s controlla i file dati con l'opzione 'thesaurus'
s{tsr} controlla il file {tsr}. Si possono specificare diverse
opzioni "s", anche parametrizzate.
i controlla i file correnti e inclusi
d controlla i file correnti e inclusi per nome o macro definiti
|i_CTRL-X_CTRL-D|
] completamento tag
t come "]"

I buffer non caricati non vengono caricati, quindi i loro autocomandi
|:autocmd| non sono eseguiti, questo potrebbe portare a completamenti
inaspettati per alcuni file (i file gzippati ad es.). I buffer
non caricati non sono controllati per il completamento di intere
righe.

Il default è ".,w,b,u,t,i", il che significa controllare:
1. il buffer corrente
2. i buffer in altre finestre
3. altri buffer caricati
4. buffer non caricati
5. tag
6. file inclusi

Come si può vedere, CTRL-N e CTRL-P possono essere usati per fare
qualsiasi espansione basata su 'iskeyword' (ad es., dizionari
|i_CTRL-X_CTRL-K|, modelli inclusi |i_CTRL-X_CTRL-I|, tag
|i_CTRL-X_CTRL-]| ed espansioni normali)

'completefunc' *'cfu'*
'completefunc' 'cfu' stringa (default: "")
locale al buffer
{non in Vi}
{non disponibile se compilato senza le funzionalità
|eval| o |+insert_expand|}
Quest'opzione specifica una funzione da usare per il completamento in
modo Insert con CTRL-X CTRL-U. |i_CTRL-X_CTRL-U|
Vedere |complete-functions| per una spiegazione di come la funzione va
richiamata, e di che cosa dovrebbe fornire.
Quest'opzione non può essere impostata da |modeline|, o nel |sandbox|,
per motivi di sicurezza.

'completeopt' *'cot'*
'completeopt' 'cot' stringa (default: "menu,preview")
globale
{non disponibile se compilato senza la funzionalità

|**eval**| o |**+insert_expand**|}
{non in Vi}

Una lista di opzioni, separate da virgola, per il completamento in modo Insert |**ins-completion**|. I valori supportati sono:

menu Usare un menù dinamico (popup menu) per visualizzare i possibili completamenti. Il menù è visualizzato solo quando esiste più di una corrispondenza, e sono disponibili colori a sufficienza. |**ins-completion-menu**|

menuone Usare un menù dinamico anche quando c'è un'unica corrispondenza. Utile se viene visualizzata qualche informazione ulteriore riguardo alla corrispondenza, ad es. da quale file proviene.

longest Inserisci soltanto la parte di testo comune a tutte le corrispondenze trovate. Se il menù è visualizzato, si può usare CTRL-L per aggiungere altri caratteri. Se si debba ignorare maiuscolo/minuscolo dipende dal tipo di completamento. Per testo contenuto in un buffer, viene usata l'opzione 'ignorecase'.

preview Mostra informazioni ulteriori riguardo al completamento scelto, nella finestra di anteprima. Ha effetto solo se combinato con "menu" o "menuone".

noinsert Non inserire alcun testo come corripondente, se l'utente a match from the menu. Only works in combination with non sceglie una corrispondenza dal menù. Ha effetto solo se combinato con "menu" o "menuone". Non ha effetto se è presente "longest".

noselect Non scegliere una corrispondenza nel menù, ma obbliga l'utente a sceglierne una dal menù. Ha effetto solo se combinato con "menu" o "menuone".

'concealcursor' *'cocu'*
'concealcursor' 'cocu' stringa (default: "")
locale alla finestra
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+conceal**|}

Imposta i Modi in cui il testo può essere nascosto ANCHE nella linea in cui è presente il cursore. Quando il Modo in cui si sta lavorando è compreso nella lista, allora il testo viene nascosto proprio come nella altre linee (in cui non c'è il cursore).

n	Modo Normal
v	Modo Visual
i	Modo Insert
c	Modo edit Linea-Comandi, per 'incsearch'

Con 'v' si indicano tutte le linee nell'area Visual, non solo quella che contiene il cursore.
Un valore utile è "nc". È usato nei file di help. Finché state scorrendo il file, il testo è nascosto, ma se iniziate a inserire del testo o se selezionate un'area in Modo Visual, il testo nascosto è visualizzato, per consentirvi di vedere cosa state facendo.
Tenete presente che la posizione del cursore non sempre è dove il cursore è visibile. Per esempio, se ci si sposta in verticale, il cursore può spostarsi su un'altra colonna.

'conceallevel' *'cole'*
'conceallevel' 'cole' numero (default: 0)
locale alla finestra
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+conceal**|}

Determina come visualizzare il testo sintatticamente marcato come "conceal" |**:syn-conceal**|:

Valore	Effetto ~
0	Il testo è visualizzato normalmente
1	Ogni blocco di testo nascosto è rimpiazzato con un

carattere. Se per l'elemento sintattico non è stato definito un carattere di rimpiazzo (vedere |:syn-cchar|) si usa il carattere definito in 'listchars' (il default è uno spazio). È evidenziato con il gruppo di evidenzianion "Conceal".

2 Il testo nascosto resta completamente invisibile a meno che sia stato definito un carattere di rimpiazzo personalizzato (vedere |:syn-cchar|).

3 Il testo nascosto è completamente invisibile.

Nota: nella linea del cursore il testo nascosto rimane visibile, per consentire la modifica e la copia del testo. Questo comportamento è modificabile tramite l'opzione 'concealcursor'.

'confirm' *'cf'* *'noconfirm'* *'nocf'*

'confirm' 'cf' booleana (default: off)
globale
{non in Vi}

Quando 'confirm' è a on, certe operazioni che normalmente fallirebbero a causa di cambiamenti non salvati a un buffer, ad es. ":q" e ":e", fanno apparire invece un dialogo |**dialog**| per chiedere se si vuole salvare il file corrente. Si può comunque usare un ! per non riscrivere |**abandon**| un buffer.
Se 'confirm' è a off si può ancora attivare la conferma per un solo comando (molto utile nelle mappature) con il comando |**:confirm**|.
Vedere anche la funzione |**confirm()**| e l'opzione 'v' in 'guioptions'.

'conskey' *'consk'* *'noconskey'* *'noconsk'*

'conskey' 'consk' booleana (default: off)
globale
{non in Vi} {solo per MS-DOS}

Quest'opzione riguardava MS-DOS e non è più supportata.

'copyindent' *'ci'* *'nocopyindent'* *'noci'*

'copyindent' 'ci' booleana (default: off)
locale al buffer
{non in Vi}

Copia la struttura della righe esistenti quando si autoindenta una nuova riga. Normalmente la nuova indentatura è ricreata con una serie di caratteri tab seguiti dal numero di spazi richiesto (a meno che |**'expandtab'**| sia abilitato, nel qual caso si usano solo spazi). L'impostazione di quest'opzione fa utilizzare per la nuova riga una copia di qualsiasi carattere sia stato usato per l'indentatura della riga esistente. 'expandtab' non ha effetto su questo caratteri, una <Tab> resta una <Tab>. Se la nuova indentatura è maggiore di quella della riga copiata, lo spazio rimanente è riempito in maniera normale.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.
Vedere anche 'preserveindent'.

'cpoptions' *'cpo'* *cpo*

'cpoptions' 'cpo' stringa (default Vim: "aABceFs",
default Vi: tutti i flag)
globale
{non in Vi}

Una sequenza di flag ognuno di un carattere. Quando un carattere è present, sta ad indicare un comportamento compatibile con Vi. Da usare quando si desidera per lo più o talora un comportamento non compatibile con Vi.
'cpoptions' sta per "compatible-options" [opzioni compatibili].
Si possono aggiungere virgole per una migliore leggibilità.
Per evitare problemi con flag che possono venire aggiunti in futuro, usare la notazione "+=" e "-=" di ":set" |**add-option-flags**|.
NOTA: Quest'opzione è impostata al valore di default di Vi quando 'compatible' è a on oppure al valore di default di Vim quando 'compatible' è a off.
NOTA: Quest'opzione è impostata al valore di default POSIX alla partenza, nel caso in cui verrebbe usato il valore di default di Vi e se esiste la variabile d'ambiente $VIM_POSIX |**posix**|. Ciò vuol dire che Vim tenta di comportarsi come previsto nella specifica POSIX.

contiene comportamento ~

cpo-a

a Se inclusa, un comando ":read" avente come argomento un nome di file imposterà il nome di file alternato per la finestra corrente.

cpo-A

A Se inclusa, un comando ":write" avente come argomento un nome di file imposterà il nome di file alternato per la finestra corrente.

cpo-b

b "\|" in un comando ":map" è riconosciuto come delimitatore [fine] del comando map. Il '\' è incluso nella mappatura, il testo dopo il '|' è interpretato come il comando seguente. Usare un CTRL-V invece che un backslash per includere il '|' nella mappatura. Usato per tutte le mappature, abbreviazioni, menù ed autocomandi.
Vedere anche |**map_bar**|.

cpo-B

B Un backslash non ha un significato speciale nelle mappature, abbreviazioni e nella parte destra dei menù di comandi.
Mettete a off questo flag per poter usare un backslash come si usa un CTRL-V. ad es., il comando ":map X \<Esc>" fa sì che X sia mappato come:
'B' inclusa: "\^[" (^[è un <Esc> vero)
'B' esclusa: "<Esc>" (5 caratteri)
('<' [descritto sotto] viene escluso in entrambi i casi)

cpo-c

c La ricerca continua alla fine di ogni corrispondenza trovata alla posizione del cursore ma non oltre l'inizio della riga seguente. Quando non è presente, la ricerca continua dal carattere DOPO la posizione del cursore. Con 'c' "abababababab" trova solo tre corrispondenze quando si ricerca "/abab", senza 'c', le corrispondenze sono cinque.

cpo-C

C Non concatenare righe di comandi eseguiti che iniziano con backslash. Vedere |**line-continuation**|.

cpo-d

d Usare "./" nell'opzione 'tags' non chiede di usare il file di tag relativamente al file corrente, ma il file di tag contenuto nella directory corrente.

cpo-D

D Non si può usare CTRL-K per immettere un digramma dopo comandi in modo Normal con un argomento di tipo carattere, tipo |**r**|, |**f**| e |**t**|.

cpo-e

e Nell'eseguire un registro con ":@r", aggiungere sempre <CR> all'ultima riga, anche quando il registro non è formato da righe. Nel caso in cui questo flag non sia attivato, ed il registro non sia formato da righe, e l'ultima riga non finisca con <CR>, l'ultima riga è visualizzata nella riga dei comandi, e può essere modificata prima di immettere <CR>.

cpo-E

E È sbagliato usare "y", "d", "c", "g~", "gu" o "gU" con una regione vuota. Questi operatori funzionano solo se hanno almeno un carattere su cui operare. Ad esempio: Questo flag fa fallire "y0" se battuto sulla prima colonna.

cpo-f

f Se inclusa, un comando ":read" avente come argomento un nome di file imposterà il nome di file per il buffer corrente, se questo ancora non ha associato un nome di file.

cpo-F

F Se inclusa, un comando ":write" avente come argomento un nome di file imposterà il nome di file per il buffer corrente, se questo ancora non ha associato un nome di file. Vedere anche |**cpo-P**|.

cpo-g

g Vai alla riga 1 se si usa ":edit" senza argomento.

cpo-H

H Quando si usa "I" su una riga che contiene solo spazi bianchi, inserire prima dell'ultimo spazio bianco. Senza questo flag, inserire dopo l'ultimo spazio bianco.

cpo-i

i Se inclusa, l'interruzione della lettura di un file lo farà marcare come modificato.

cpo-I

I Quando si nuove il cursore in su o in giù subito dopo aver inserito un'indentatura per 'autoindent', non cancellare l'indentatura.

cpo-j

j Quando si uniscono righe, aggiungere due spazi solo dopo '.', e non dopo '!' o '?'. Vedere anche 'joinspaces'.

cpo-J

J Una frase |**sentence**| deve essere seguita da due spazi dopo il '.', '!' o '?'. Un <Tab> non è riconosciuto equivalere a uno spazio bianco.

cpo-k

k Disabilita il riconoscimento di codici di tasto "grezzi" nelle mappature, abbreviazioni, e nella parte destra dei comandi di menù. Ad esempio, se un tasto <Key> invia ^[OA (dove ^[è <Esc>), il comando ":map X ^[OA" risulta nella mappatura di X come:

```
        'k' inclusa:    "^[OA"   (3 caratteri)
        'k' esclusa:    "<Key>"  (un tasto)
```

Vedere anche il flag '<' più sotto.

cpo-K

K Non aspettare che un codice di tasto si completi quando è a metà di una mappatura. Questo inibisce una mappatura <F1><F1> dopo che solo una parte del secondo <F1> è stata letta. È così possibile cancellare la mappatura battendo <F1><Esc>.

cpo-l

l Backslash in un'espressione di ricerca di un elenco espresso entro [] viene interpretato letteralmente, solo "\]", "\^", "\-" e "\\" sono speciali.
Vedere |**/[]**|

```
   'l' inclusa: "/[ \t]"  trova <Spazio>, '\' e 't'
   'l' esclusa: "/[ \t]"  trova <Spazio> e <Tab>
```

Vedere anche |**cpo-**|.

cpo-L

L Se l'opzione 'list' è attiva, 'wrapmargin', 'textwidth', 'softabstop' e il modo Virtual Replace (vedere |**gR**|) contano un <Tab> come due caratteri, invece di comportarsi come si fa normalmente con il <Tab>.

cpo-m

m Se inclusa, l'opzione showmatch attende sempre per un mezzo secondo. Se esclusa, l'opzione showmatch aspetta mezzo secondo, o l'immissione di un carattere.
|**'showmatch'**|

cpo-M

M Se esclusa, la ricerca di corrispondenze del comando "%" terrà conto dei backslash. Quindi in "(\( )" e "\((\)" le parentesi esterne si corrispondono.
Se inclusa, "%" ignora i backslash, per compatibilità con Vi.

cpo-n

n Se inclusa, la colonna usata per visualizzare il numero riga 'number' e quello relativo 'relativenumber' sarà anche usato per contenere il testo di righe che vanno a capo perché più lunghe dell'ampiezza dello schermo.

cpo-o

o La posizione nella riga per un comando di ricerca non è ricordata in una ricerca successiva.

cpo-O

O Niente lamentele se un file viene sovrascritto, anche se non esisteva quando si è cominciato ad editarlo. Questa è una protezione di un file che sia stato

creato imprevedibilmente da qualcun altro. Via non si lamentava in questo caso.

cpo-p

p indentatura Lisp compatibile con Vi. Se non presente, viene usato un algoritmo un filo migliore.

cpo-P

P Quando inclusa, un comando ":write" che aggiunga in fondo a un file imposterà il nome di quel file come nome per il buffer corrente, se il buffer corrente ancora non ha ancora un nome di file associato e il flag 'F' è stato pure incluso |**cpo-F**|.

cpo-q

q Quando si uniscono (comando "join") molte righe, lasciare il cursore alla posizione in cui si troverebbe se si stessero riunendo solo due righe.

cpo-r

r Il comando "riesegui-l'ultimo-comando" (".") usa "/" per ripetere un comando di ricerca, invece di utilizzare di nuovo la stringa di ricerca immessa.

cpo-R

R Rimuove marcature dalle righe filtrate. Senza questo flag le marcature sono mantenute, come se si fosse usato il comando |**:keepmarks**|.

cpo-s

s Imposta opzioni di un buffer quando si entra nel buffer per la prima volta. Questo comportamento è quello del Vim 3.0. Ed è anche il valore di default. Se manca, le opzioni sono impostate alla creazione del buffer.

cpo-S

S Imposta sempre opzioni di un buffer quando si entra nel buffer (tranne 'readonly', 'fileformat', 'filetype' e 'syntax'). Questa impostazione è la (più) compatibile con Vi.
Le opzioni sono impostate coi valori del buffer corrente. Se cambiate un'opzione e andate in un altro buffer, il valore è copiato. In effetti rende le opzioni di buffer globali, ossia valide per ogni buffer.

's'	'S'	copiatura opzioni di buffer
no	no	alla creazione del buffer
yes	no	alla prima entrata nel buffer (default)
X	yes	ogni volta che si entra nel buffer (comportamento compatibile con Vi)

cpo-t

t L'espressione di ricerca del comando tag è ricordata per un successivo uso col comando "n". Altrimenti Vim mette l'espressione nella storia delle espressioni cercate, ma non cambia l'ultima espressione di ricerca utilizzata.

cpo-u

u Undo compatibile con Vi. Vedere |**undo-two-ways**|.

cpo-v

v Caratteri ricoperti col tasto backspace restano visibili sullo schermo mentre si è in modo Insert. Quando questo flag non è attivato i caratteri sono subito cancellati dallo schermo. Con questo flag il testo nuovamente immesso si sovrappone ai caratteri precedentemente ricoperti.

cpo-w

w Se si usa "cw" su un carattere bianco, cambiare solo quel carattere, e non tutti i caratteri bianchi fino all'inizio della parola seguente.

cpo-W

W Non sovrascrivere un file in modalità di sola lettura. Se omesso, ":w!" sovrascrive un file in sola lettura, quando ciò sia possibile.

cpo-x

x <Esc> sulla riga-comandi esegue la riga-comandi. Il default in Vim è di ignorare la riga-comandi, poiché <Esc> normalmente abortisce il comando. |**c_<Esc>**|

cpo-X

X Se si usa un contatore con "R" il testo sostituito è cancellato solo una volta. Lo stesso succede quando si ripete "R" usando il comando "." e un contatore.

cpo-y

y Un comando yank può essere ripetuto con ".".

cpo-Z

Z Quando si usa "w!" mentre è impostata l'opzione 'readonly', non annullare 'readonly'.

cpo-!

! Nel rifare un comando di filtro, usa il comando esterno usato per ultimo, qualunque fosse. Altrimenti viene usato l'ultimo comando di -filtro-.

cpo-$

$ Mentre si modifica una riga, non visualizzare la riga, ma mettere un '$' alla fine del testo da modificare. Il testo da modificare verrà sovrascritto quando si batte il nuovo testo. La riga è visualizzata nuovamente dando dei comandi che spostano il cursore dal punto di inserimento.

cpo-%

% La corrispondenza per il comando "%" è fatta in maniera compatibile con Vi.
"#if", "#endif", etc. non sono riconosciuti.
"/*" e "*/" non sono riconosciuti.
Parentesi all'interno di [stringhe delimitate da] apici singoli e doppi sono contate, così che una stringa contenente una parentesi disturba la corrispondenza. Ad esempio, in una riga come "if (strcmp("foo(", s))" la prima parentesi non corrisponde all'ultima. Quando questo flag non è attivata, parentesi contenute all'interno di apici singoli e doppi sono trattate in maniera speciale.
Quando si cerca una corrispondenza ad una parentesi inserita all'interno di apici, ne verrà trovata una corrispondente (se ne esiste una).
Ciò funziona molto bene per programmi C.
Questo flag è anche usato per altre funzionalità di Vim, come l'indentatura nel linguaggio C.

cpo--

- Se inclusa, un comando di movimento verticale non viene eseguito quando andrebbe sopra la prima riga, oppure sotto l'ultima riga. Se non inclusa, il cursore si sposta alla prima o all'ultima riga, tranne nel caso in cui già si trovi lì.
Si applica ai comandi "-", "k", CTRL-P, "+", "j", CTRL-N, CTRL-J e ":1234".

cpo-+

+ Se inclusa, un comando ":write file" metterà a off il flag 'modified' del buffer, anche se il buffer stesso può ancora essere differente dal file di partenza.

cpo-star

* Usare ":*" come ":@". Quando questo flag non è attivata ":*" è un alias per ":'<,'>", seleziona l'area Visuale.

cpo-<

< Disabilita il riconoscimento di tasti indicati con la notazione |**<>**| in mappature, abbreviazioni, e nella parte destra dei comandi menù. Ad esempio, il comando ":map X <Tab>" risulta nel mappare X come:
'<' inclusa: "<Tab>" (5 caratteri)
'<' esclusa: "^I" (^I è il tasto <Tab>)
Vedere anche il flag 'k' più sopra.

cpo->

> Quando si aggiunge qualcosa in fondo a un registro, inserire un "a capo" prima del testo aggiunto.

cpo-;

; Quando si usa |**,**| oppure |**;**| per ripetere l'ultima ricerca |**t**| e il cursore si trova proprio davanti al carattere cercato, il cursore non si sposta.
Se non si specifica questo flag, il cursore

salterà il carattere in questione e si posizionerà
sulla successiva occorrenza del carattere stesso.

Flag POSIX. Questi flag non sono inseriti nel valore di default di
Vi, tranne quando sia stato impostato alla partenza $VIM_POSIX |**posix**|

contiene	comportamento ~	
		cpo-#
#	Un contatore prima di "D", "o" e "O" viene ignorato.	
		cpo-&
&	Se si è usato il comando ":preserve", mantenere il file di swap quando si esce normalmente mentre il buffer è ancora in memoria. Questo flag è controllato al momento di uscire da Vim.	
		cpo-
\	Backslash (\]) messo tra [] in un'espressione di ricerca è preso letteralmente, solo "\]" è speciale. Vedere |**/[]**| '\' incluso: "/[\-]" trova <Spazio>, '\' e '-' '\' escluso: "/[\-]" trova <Spazio> e '-' vedere anche |**cpo-l**|.	
		cpo-/
/	Quando si usa "%" come stringa da sostituire nel comando |**:s**|, usa la stringa di sostituzione usata per ultima. |**:s%**|	
		cpo-{
{	I comandi |**{**| e |**}**| si fermano anche a un carattere "{" che si trova all'inizio di una riga.	
		cpo-.
.	I comandi ":chdir" e ":cd" non vengono eseguiti se il buffer corrente è modificato, a meno di aggiungere ! . Vim non ne ha bisogno, in quanto ricorda il nome completo di un file aperto.	
		cpo-bar
|	I valori delle variabili d'ambiente $LINES e $COLUMNS prevalgono sui valori delle dimensioni del terminale ottenute richiamando funzioni specifiche di sistema.	

'cryptmethod' *'cm'*

'cryptmethod' 'cm' stringa (default: "zip")
globale o locale al buffer |**global-local**|
{non in Vi}

Metodo usato per cifratura quando il buffer è scritto su un file:

pkzip

zip Metodo compatibile con PkZip. Un tipo di cifratura "leggera". Compatibile all'indietro con Vim 7.2 e versioni precedenti.

blowfish

blowfish Metodo Blowfish. Cifratura mediamente forte. Ma con un difetto di implementazione. Richiede Vim 7.3 o successivi, i file NON sono leggibili con vim 7.2 o versioni precedenti.
Il metodo aggiunge ogni volta al file un "seme" (seed) iniziale, per cui ogni volta che si riscrive il file, i byte cifrati saranno completamente differenti [anche se il file è riscritto senza essere stato modificato -NdT].

blowfish2 Metodo Blowfish. Cifratura mediamente forte. Richiede Vim 7.4.401 o successivi, i file NON sono leggibili con vim 7.3 o versioni precedenti.
Il metodo aggiunge ogni volta al file un "seme" (seed) iniziale, per cui ogni volta che si riscrive il file, i byte cifrati saranno completamente differenti [anche se il file è riscritto senza essere stato modificato -NdT]. L'intero file di undo è cifrato, non solo le porzioni contenenti del testo.

Si dovrebbe usare il metodo "blowfish2", anche per cifrare nuovamente file che erano già stati cifrati in precedenza [con altri metodi].

Quando si legge un file cifrato, 'cryptmethod' verrà impostato automaticamente al metodo utilizzato sul file letto. Quindi se lo si

riscrive senza cambiare 'cryptmethod' verrà utilizzato lo stesso
metodo.
Cambiare 'cryptmethod' non fa sì che il file venga considerato
modificato. Perché la nuova cifratura abbia effetto dovete chiedere
che il file sia riscritto, e non viene inviato alcun avviso (per file
modificato e non ancora salvato) a meno che non siano state effettuate
altre modifiche. Vedere anche |**:X**|.

Quando il valore globale non è impostato (stringa nulla), verrà
utilizzato per default il valore "zip". Quando si imposta come valore
locale la stringa nulla, verrà utilizzato il valore globale.

Qualora un nuovo metodo di cifratura venga aggiunto in una futura
versione di Vim e quella corrente non lo riconosca, la cosa verrà
segnalata con il messaggio *E821* . In questo caso dovete andare a
modificare quel file con una versione più recente di Vim.

'cscopepathcomp' *'cspc'*
'cscopepathcomp' 'cspc' numero (default: 0)
globale
{non disponibile se compilato senza la funzionalità
|**+cscope**|}
{non in Vi}
Determina quante componenti del nome completo di file visualizzare in
una lista di tag. Vedere |**cscopepathcomp**|.
NOTA: Quest'opzione è impostata a 0 se si attiva 'compatible'.

'cscopeprg' *'csprg'*
'cscopeprg' 'csprg' stringa (default: "cscope")
globale
{non disponibile se compilato senza la funzionalità
|**+cscope**|}
{non in Vi}
Comando da usare per eseguire cscope. Vedere |**cscopeprg**|.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|,
per motivi di sicurezza.

'cscopequickfix' *'csqf'*
'cscopequickfix' 'csqf' stringa (default: "")
globale
{non disponibile se compilato senza la funzionalità
|**+cscope**| o |**+quickfix**|}
{non in Vi}
Specifica se usare la finestra quickfix per mostrare i risultati di
cscope. Vedere |**cscopequickfix**|.

'cscoperelative' *'csre'* *'nocscoperelative'* *'nocsre'*
'cscoperelative' 'csre' booleana (default: off)
globale
{non disponibile se compilato senza la funzionalità
|**+cscope**|}
{non in Vi}
In mancanza di un prefisso (-P) per cscope, impostando quest'opzione
si chiede di usare come prefisso il nome di directory del file
cscope.out.
Vedere |**cscoperelative**|.

'cscopetag' *'cst'* *'nocscopetag'* *'nocst'*
'cscopetag' 'cst' booleana (default: off)
globale
{non disponibile se compilato senza la funzionalità
|**+cscope**|}
{non in Vi}
Usare cscope per comandi tag. Vedere |**cscope-options**|.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'cscopetagorder' *'csto'*
'cscopetagorder' 'csto' numero (default: 0)
globale
{non disponibile se compilato senza la funzionalità
|**+cscope**|}
{non in Vi}

Determina l'ordine in cui ":cstag" esegue una ricerca. Vedere
|**cscopetagorder**|.
NOTA: Quest'opzione è impostata a 0 se si attiva 'compatible'.

'cscopeverbose' *'csverb'*
'nocscopeverbose' *'nocsverb'*
'cscopeverbose' 'csverb' Booleana (default: off)
globale
{non disponibile se compilato senza la funzionalità
|**+cscope**|}
{non in Vi}
Informa quando si aggiunge un database cscope.
Vedere |**cscopeverbose**|.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'cursorbind' *'crb'* *'nocursorbind'* *'nocrb'*
'cursorbind' 'crb' booleana (default: off)
locale alla finestra
{non in Vi}
Se quest'opzione è impostata, quando il cursore nella finestra
corrente si sposta, le altre finestre col cursore collegato (finestre
che hanno anche loro impostato la stessa opzione) spostano i loro
cursori alla riga e colonna corrispondetne. Quest'opzione è utile per
visualizzare le diffrenze fra due versioni di un file (vedere 'diff');
in modo Diff, le righe inserite e cancellate (ma non i caratteri
all'interno di una singola riga) sono tenuti in considerazione.

'cursorcolumn' *'cuc'* *'nocursorcolumn'* *'nocuc'*
'cursorcolumn' 'cuc' booleana (default: off)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+syntax**|}
Evidenzia la colonna del cursore sullo schermo con l'evidenziazione
CursorColumn |**hl-CursorColumn**|. Utile per allineare del testo.
Rallenta il rinfresco della videata.
Se desiderate l'evidenziazione solo nella finestra corrente, è
possibile usare i seguenti autocomandi: >
au WinLeave * set nocursorline nocursorcolumn
au WinEnter * set cursorline cursorcolumn
<
'cursorline' *'cul'* *'nocursorline'* *'nocul'*
'cursorline' 'cul' booleana (default: off)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+syntax**|}
Evidenzia la riga del cursore sullo schermo con l'evidenziazione
CursorLine |**hl-CursorLine**|. Utile per visualizzare facilmente il
cursore. Rallenta il rinfresco della videata.
Quando è attivo il modo Visual, l'evidenziazione non è usata, per
rendere più facile l'individuazione del testo selezionato.

'debug'
'debug' stringa (default: "")
globale
{non in Vi}
Si possono usare questi valori:
msg Messaggi di errore che verrebbero altrimenti omessi sono
visualizzati in ogni caso.
throw Messaggi di errore che verrebbero altrimenti omessi sono
visualizzati in ogni caso, e inoltre generano un'eccezione
e impostano |**v:errmsg**|.
beep Viene dato un messaggio in casi in cui normalmente viene dato
solo un "beep".
I valoro possono essere combinati fra loro, separati da virgola.
"msg" e "throw" sono utili nel debug di 'foldexpr', 'formatexpr' o
'indentexpr'.

'define' *'def'*
'define' 'def' stringa (default: "^\s*#\s*define")
globale o locale al buffer |**global-local**|
{non in Vi}

Espressione da usare per trovare una definizione macro. È una espressione di ricerca come per il comando "/". Quest'opzione è usata per comandi come "[i" e "[d" |**include-search**|. L'opzione 'isident' è usata per riconoscere i nomi definiti dopo aver trovato la corrispondenza:
{corr. con 'define'}{carat. non-ID}{nome definito}{carat. non-ID}
Vedere |**option-backslash**| per inserire spazi e backslash.
Il valore di default è per programmi C. Per C++ può essere utile il valore seguente, per includere dichiarazioni di tipi di costanti: >
^\(#\s*define\|[a-z]*\s*const\s*[a-z]*\)
< Immettendo il comando ":set", i backslash vanno raddoppiati!

'delcombine' *'deco'* *'nodelcombine'* *'nodeco'*
'delcombine' 'deco' booleana (default: off)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+multi_byte**|}
Se editate Unicode e quest'opzione è impostata, backspace e 'x' in modo Normal cancellano ogni singolo carattere composto. Quando l'opzione è a off (questo è il default), il carattere e quelli di cui esso fa parte sono cancellati.
Nota: Quando 'delcombine' è impostato "xx" può dare risultati diversi da "2x"!

Questo è utile per Arabo, Ebraico e molte altre lingue in cui ci possono essere caratteri composti assieme a caratteri base, se si desidera rimuovere solo quelli composti.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'dictionary' *'dict'*
'dictionary' 'dict' stringa (default: "")
globale o locale al buffer |**global-local**|
{non in Vi}
Lista di nomi file, separati da virgola, usati per ottenere parole ad uso dei comandi di completamento parole chiave |**i_CTRL-X_CTRL-K**|.
Ogni file dovrebbe contenere una lista di parole. Ci può essere una parola per riga, o più parole su ogni riga, separate da un carattere che non si faccia parte delle parole (meglio se uno spazio bianco).
La lunghezza massima di ogni riga è di 510 byte.

Quando quest'opzione è nulla, o è presente un elemento "spell", e la correzione ortografica è abilitata, le parole contenute nella lista di parole del linguaggio 'spelllang' correntemente specificato sono utilizzate. Vedere |**spell**|.

Per inserire una virgola in un nome file, metteteci davanti un backslash. Vedere |**option-backslash**| per l'uso dei backslash.
Quest'opzione non ha niente a che vedere con il tipo di variabile |**Dictionary**|.
Dove trovare una lista di parole?
- In FreeBSD, c'è il file "/usr/share/dict/words".
- Nell'archivio Simtel, vedere la directory "msdos/linguist".
- Nella collezione GNU, vedere "miscfiles".
L'uso di |**:set+=**| e |**:set-=**| è preferito per aggiungere o togliere directory dalla lista. Questo evita problemi quando una versione futura usasse un default differente.
Comandi contenuti fra "`" (backtick) non sono permessi in questa opzione per motivi di sicurezza.

'diff' *'nodiff'*
'diff' booleana (default: off)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+diff**|}
Aggiunge la finestra corrente al gruppo di finestre che mostrano le differenze fra file. Vedere |**vimdiff**|.

'dex' *'diffexpr'*
'diffexpr' 'dex' stringa (default: "")
globale
{non in Vi}

{non disponibile se compilato senza la funzionalità
|**+diff**|}
Espressione da valutare per ottenere un file che descriva le
differenze fra due versioni di un file nello stile di "ed" (l'editor
line-mode). Vedere |**diff-diffexpr**|.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|,
per motivi di sicurezza.

'dip' *'diffopt'*
'diffopt' 'dip' stringa (default: "filler")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+diff**|}
Impostazione opzioni per modo Diff. Può contenere gli elementi che
seguono. Nessuna è obbligatoria.
Gli elementi sono separati da una virgola.

filler	Mostra righe di riempimento, per tenere in sincronia il testo con una finestra che contenga altre righe nella stessa posizione. Utile soprattutto quando le finestre sono fianco a fianco e 'scrollbind' è impostato.
context:{n}	Mostra un contesto di {n} righe tra una differenza ed la successiva piegatura che contiene righe non modificate. Se omesso, si usa un contesto di sei righe. Vedere |**fold-diff**|.
icase	Ignora cambi maiuscole/minuscole in un testo. "a" e "A" sono considerati uguali. Aggiunge il flag "-i" al comando "diff" se 'diffexpr' non è specificata.
iwhite	Ignora cambi solo nel numero di spazi bianchi. Aggiunge il flag "-b" al comando "diff" se 'diffexpr' non è specificata. Controllate la documentazione del comando "diff" per gli effetti esatti. Spazi bianchi in fondo alla riga dovrebbero venire ignorati, ma non quelli a inizio riga.
horizontal	Inizia il modo Diff spezzando lo schermo in orizzontale, a meno che ci sia una richiesta specifica differente.
vertical	Inizia il modo Diff spezzando lo schermo in verticale, a meno che ci sia una richiesta specifica differente.
hiddenoff	Non usa il modo Diff per un buffer, quando diviene nascosto.
foldcolumn:{n}	Imposta l'opzione 'foldcolumn' a {n} quando si entra nel modo Diff. In assenza di specifiche, è usato il valore 2.

Ad es.: >

```
:set diffopt=filler,context:4
:set diffopt=
:set diffopt=filler,foldcolumn:3
<
```

'digraph' *'dg'* *'nodigraph'* *'nodg'*
'digraph' 'dg' booleana (default: off)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+digraphs**|}
Permette l'inserimento di digrammi in modo Insert battendo {car1} <BS>
{car2}. Vedere |**digraphs**|.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'directory' *'dir'*
'directory' 'dir'	stringa (default per Amiga: ".,t:",
per MS-DOS e Win32: ".,$TEMP,c:\tmp,c:\temp"
per Unix: ".,~/tmp,/var/tmp,/tmp")
globale
Lista di nomi directory per posizionarci il file di swap, separate da virgole.
- Il file di swap sarà creato nella prima directory in cui risulta possibile farlo.
- Una lista vuota preclude l'utilizzo di file di swap (il ripristino è per conseguenza impossibile!).
- Una directory "." chiede di mettere il file di swap nella stessa directory del file in modifica. In Unix, un punto è anteposto al nome del file, così che non sia visualizzato nella lista della directory. In MS-Windows si imposta l'attributo "hidden" (nascosto) e lo si fa precedere da un ".", quando possibile.
- Una directory il cui nome inizi con "./" (o ".\" per MS-DOS et al.) chiede di mettere il file di swap in una directory relativamente a quella in cui si trova il file in modifica. Il "." iniziale è sostituito col nome della directory del file in modifica.
- Per Unix e Win32, se una directory finisce con due delimitatori di nome file "//" o "\\", il nome del file di swap verrà costruito col nome completo del percorso che conduce al file con tutti i delimitatori di nome file sostituiti dal segno '%'. Questo serve a garantire l'unicità del nome del file nella directory di salvaguardia.
In ambiente Win32, quando una virgola di separazione segue, si deve usare "//", poiché "\\" includerebbe la virgola nel nome del file.
- Spazi dopo la virgola sono ignorati, gli altri spazi sono considerati come facenti parte del nome della directory. Per avere uno spazio all'inizio di un nome di directory, premettetegli un backslash [\].
- Per includere una virgola in un nome di directory, premettetegli un backslash [\].
- Un nome di directory può terminare con un ':' o '/'.
- Le variabili d'ambiente vengono valutate |**:set_env**|.
- Attenti ai caratteri '\', mettetene uno davanti a uno spazio, mettetene due per ottenerne uno nell'opzione (vedere |**option-backslash**|), ad es.: >
```
:set dir=c:\\tmp,\ dir\\,with\\,commas,\\\ dir\ with\ spaces
```
< - Per compatibilità all'indietro con **Vim version 3.0** un '>' all'inizio dell'opzione viene rimosso.
L'uso del "." come primo valore nella lista è raccomandato. Ciò fa sì che si riceva un avvertimento andando due volte in modifica sullo stesso file. L'uso di "/tmp" sotto Unix è sconsigliato: se il sistema cade, voi perdete il file di swap.
"/var/tmp" spesso non viene "pulito" alla ripartenza del sistema, e quindi è una scelta migliore di "/tmp". Ma può contenere molti file, e il vostro file di swap si perde nella folla. Ecco perché viene considerata prima una directory "tmp" nella vostra home directory.
L'uso di |**:set+=**| e |**:set-=**| è preferito per aggiungere o togliere directory dalla lista. Ciò permette di evitare problemi laddove una futura versione usasse un altro default.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.
{Vi: directory per mettere file temporanei, valore di default "/tmp"}

'display' *'dy'*
'display' 'dy'	stringa (default: "", impostata a "truncate"in
|**defaults.vim**|)
globale
{non in Vi}
Cambia il modo in cui il testo è visualizzato. Lista di flag, separate da virgola:
lastline	Se inclusa, quanto è possibile visualizzare nell'ultima riga di una finestra viene visualizzato. La stringa "@@@" è posta in fondo all'ultima riga visualizzata sullo schermo, per indicre che il resto della riga non è visualizzato.
truncate	Come "lastline", ma la stringa "@@@" è visualizzata nella prima colonna dell'ultima riga visibile sullo schermo. Prevale su "lastline".

```
        uhex            Mostra caratteri non stampabili in esadecimale <xx>,
                        invece che usare ^C e ~C.

        Se né "lastline" né "truncate" sono stati specificati, un'ultima riga
                        che non possa essere visualizzata completamente sullo
                        schermo è sostituita da righe contenenti solo il
                        carattere "@".

                                                *'eadirection'* *'ead'*
'eadirection' 'ead'     stringa (default: "both")
                        globale
                        {non in Vi}
                        {non disponibile se compilato senza la funzionalità
                        |+vertsplit|}
        Specifica quando è applicabile l'opzione 'equalalways':
                ver     verticalmente, larghezza della finestra invariata
                hor     orizzontalmente, altezza della finestra invariata
                both    variano sia larghezza che altezza della finestra

                           *'ed'* *'edcompatible'* *'noed'* *'noedcompatible'*
'edcompatible' 'ed'     booleana        (default: off)
                        globale
        cambia i flag 'g' e 'c' del comando ":substitute" ogni volta che un
        flag è specificato.  Vedere |complex-change|.  Vedere anche l'opzione
        'gdefault'.
        L'attivazione di quest'opzione può rendere non funzionanti dei plugin!

                                  *'emoji'* *'emo'* *'noemoji'* *'noemo'*
'emoji' 'emo'           booleana        (default: on)
                        globale
                        {non in Vi}
                        {disponibile solo se compilato con la funzionalità
                        |+multi_byte|}
        Quando è attiva tutti i caratteri emoji (faccine) sono considerati
        essere di larghezza intera.


                                        *'encoding'* *'enc'* *E543*
'encoding' 'enc'        stringa (default: "latin1" o valore di $LANG)
                        globale
                        {disponibile solo se compilato con la funzionalità
                        |+multi_byte|}
                        {non in Vi}
        Imposta la codifica caratteri usata in Vim.  Applicabile a testo nei
        buffer, registri, stringhe nelle espressioni, testo contenuto nel file
        viminfo file, etc.  Imposta il tipo di caratteri con cui Vim può
        lavorare.  Vedere |encoding-names| per i possibili valori.

        NOTA: Cambiando quest'opzione non cambia la codifica del testo che
        Vim sta modificando.  Può così divenire non leggibile del testo
        non-ASCII.  L'opzione dovrebbe essere lasciata al suo valore
        di default, oppure impostata alla partenza di Vim.  Vedere
        |multibyte|.  Per ricaricare i menù, vedere |:menutrans|.

        Quest'opzione non può essere impostata da |modeline|.  È molto
        probabile che alteri indebitamente il testo.

        NOTA: Da GTK+ 2 in poi, si raccomanda vivamente di impostare
        'encoding' a "utf-8".  Sebbene si sia cercato con cura di supportare
        valori diversi di 'encoding', "utf-8" è la scelta naturale in questo
        ambiente ed evita di perdere tempo inutile in conversioni.  "utf-8"
        non è stato messo come valore di default per evitare un comportamento
        differente fra la versione terminale e quella GUI, e per evitare di
        cambiare la codifica dei nuovi file creati, senza preavviso (nel caso
        in cui 'fileencodings' non sia impostato).

        La codifica caratteri dei file può essere diversa da 'encoding'.
        Questo è specificato con 'fileencoding'.  La conversione è fatta con
        iconv() o seguendo quanto specificato in 'charconvert'.

        Per accertare se 'encoding' è multi-byte, si può usare: >
                if has("multi_byte_encoding")
<
```

Normalmente 'encoding' sarà uguale al vostro "locale" corrente. Sarà questo il valore di default se Vim riconosce l'impostazione del vostro ambiente. Se 'encoding' non è impostato al vostro "locale" corrente, 'termencoding' va impostato per convertire il testo immesso e quello visualizzato. Vedere |**encoding-table**|.

Quando impostate quest'opzione, innesca l'autocomando di evento |**EncodingChanged**| in modo che si possano impostare i font se necessario.

Quando l'opzione è impostata, il valore è convertito a minuscolo. Quindi potete impostarla anche con valori maiuscoli. I caratteri '_' sono convertiti a segni '-'.
Quando la codifica è riconosciuta, viene cambiata al nome standard. Ad esempio "Latin-1" diviene "latin1", "ISO_88592" diviene "iso-8859-2" e "utf8" diviene "utf-8".

Nota: "latin1" è anche in uso se non si riesce a determinare la codifica. Questo può funzionare solo se si lavora con file con la stessa codifica! Quando il set di caratteri in uso non è latin1, assicuratevi che 'fileencoding' e 'fileencodings' siano nulli. Quando è necessaria una conversione, passate a utilizzare utf-8.

Quando "unicode", "ucs-2" o "ucs-4" sono in uso, Vim internamente usa utf-8. Voi non lo notate mentre state editando, ma serve per il file viminfo |**viminfo-file**|. E Vim si aspetta che anche il terminale usi utf-8. Quindi impostando 'encoding' ad uno di questi valori, invece che ad utf-8, ha effetto solo, per codifiche usate in un file, quando 'fileencoding' sia nullo.

Quando 'encoding' è impostato ad una codifica Unicode, e 'fileencodings' non era stato ancora specificato, il valore di default per 'fileencodings' viene modificato.

'endofline' *'eol'* *'noendofline'* *'noeol'*
'endofline' 'eol' booleana (default: on)
locale al buffer
{non in Vi}

Quando si scrive un file e quest'opzione è a off e l'opzione 'binary' è a on, oppure l'opzione 'fixeol' è a off, nessun <EOL> sarà scritto dopo l'ultima riga nel file. Quest'opzione è impostata o messa a off automaticamente quando si incomincia a editare un nuovo file, a seconda se il file ha (oppure non ha) un <EOL> per l'ultima riga nel file. Normalmente non si dovrebbe impostare o mettere a off quest'opzione.
Quando 'binary' è a off, e 'fixeol' è a on, il valore non viene usato nello scrivere il file. Quando 'binary' è a on, o 'fixeol' è a off l'opzione è usata per ricordare la presenza di un <EOL> nell'ultima riga nel file, così che quando il file viene scritto, la situazione presente nel file originale possa essere mantenuta. Ma potete cambiare l'opzione se lo desiderate.

'equalalways' *'ea'* *'noequalalways'* *'noea'*
'equalalways' 'ea' booleana (default: on)
globale
{non in Vi}

Se impostata, tutte le finestre sono automaticamente rimesse alla stessa dimensione dopo aver aperto o chiuso una finestra. Ciò succede anche nel momento in cui l'opzione è attivata. Se è a off, l'apertura di una finestra nuova ridurrà la dimensione della finestra corrente e lascerà le altre come erano prima. Quando si chiude una finestra, le righe extra sono date alla finestra adiacente (come specificato dalle opzioni 'splitbelow' e 'splitright').
Quando sono presenti finestre divise sia in verticale che in orizzontale, una dimensione minima viene calcolata, e alcune finestre possono essere più grosse dello spazio a disposizione. L'opzione 'eadirection' dice in quale direzione la dimensione viene ritoccata.
Il cambiamento in altezza e larghezza di una finestra può essere evitato impostando rispettivamente 'winfixheight' 'winfixwidth'.
Se la dimensione di una finestra viene specificata al momento della creazione della stessa, le dimensioni delle finestre già presenti non sono rese uguali (è complesso da programmare, ma potrebbe divenire possibile in futuro).

'equalprg' *'ep'*
'equalprg' 'ep' stringa (default: "")
globale o locale al buffer |**global-local**|
{non in Vi}
Programma esterno da usare per il comando "=". Quando quest'opzione non è specificata si usano le funzioni interne di formattazione; si tratta di 'lisp', 'cindent' o 'indentexpr'. Se Vim è stato compilato senza capacità di formattazione interna, si usa il programma "indent". Le variabili d'ambiente vengono valutate |**:set_env**|. Vedere |**option-backslash**| per inserire spazi e backslash. Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'errorbells' *'eb'* *'noerrorbells'* *'noeb'*
'errorbells' 'eb' booleana (default: off)
globale
Segnala (col campanello o con schermo lampeggiante) messaggi di errore. Rilevante solo per messaggi di errore, il campanello verrà usato comunque per molti errori che non generano messaggi (ad es. battere <ESC> in modo Normal. Vedere 'visualbell' per impostare il campanello in modo che suoni, lampeggi, o non faccia nulla. Vedere 'belloff' per scegliere in quali particolari casi far suonare il campanello.

'errorfile' *'ef'*
'errorfile' 'ef' stringa (Amiga default: "AztecC.Err",
others: "errors.err")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+quickfix**|}
Nome del file degli errori in modo QuickFix (vedere |**:cf**|). Se si usa l'argomento "-q" invocando Vim, 'errorfile' è impostato all'argomento che segue nella riga-comando. Vedere |**-q**|. NON usato per il comando ":make". Vedere 'makeef' al riguardo. Le variabili d'ambiente vengono valutate |**:set_env**|. Vedere |**option-backslash**| per inserire spazi e backslash. Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'errorformat' *'efm'*
'errorformat' 'efm' stringa (valore di default molto lungo)
globale o locale al buffer |**global-local**|
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+quickfix**|}
Descrizione del formato per le righe nel file degli errori in formato simile a quello della funzione [C] scanf (vedere |**errorformat**|).

'esckeys' *'ek'* *'noesckeys'* *'noek'*
'esckeys' 'ek' booleana (default Vim: on, default Vi: off)
globale
{non in Vi}
Tasti funzione che iniziano con <Esc> sono riconosciuti in modo Insert. Quando quest'opzione è a off, il cursore e i tasti funzione non sono utilizzabili in modo Insert se iniziano con un <Esc>. La ragione per questo è che il singolo <Esc> è riconosciuto subito, invece che dopo un'attesa di un secondo. Invece di mettere ad off quest'opzione, potreste scegliere di cambiare i valori di 'timeoutlen' e 'ttimeoutlen'. Notate che quando 'esckeys' è a off, potete ancora mappare qualsiasi tasto, ma i tasti cursore non funzionano per default.
NOTA: Quest'opzione è impostata al valore di default di Vi si imposta 'compatible' e al valore di default di Vim quando 'compatible' viene messa a off.

'eventignore' *'ei'*
'eventignore' 'ei' stringa (default: "")
globale
{non in Vi}
Una lista di nomi di eventi, che innescano autocomandi, da ignorare. Se impostata ad "all", o quando "all" è uno degli elementi della

lista, tutti gli eventi che innescano autocomandi sono ignorati, e non
verranno eseguiti autocomandi.
Altrimenti, questa è una lista di nomi di eventi, separati da virgola.
Ad es.: >
:set ei=WinEnter,WinLeave
<

'expandtab' *'et'* *'noexpandtab'* *'noet'*
'expandtab' 'et' booleana (default: off)
locale al buffer
{non in Vi}
In modo Insert: Usa un numero adeguato di spazi per inserire un <Tab>.
Spazi vengono inseriti indentando coi comandi '>' e '<' e quando
'autoindent' è attivo. Per inserire un <Tab> con 'expandtab' attivo,
usare CTRL-V<Tab>. Vedere anche |:retab| e |ins-expandtab|.
Quest'opzione è messa a off quando si imposta l'opzione 'paste' e
ripristinata quando l'opzione 'paste' è messa a off.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'exrc' *'ex'* *'noexrc'* *'noex'*
'exrc' 'ex' booleana (default: off)
globale
{non in Vi}
Abilita la lettura di .vimrc, .exrc e .gvimrc dalla directory
corrente.

Attivare quest'opzione potrebbe rappresentare una falla nella
sicurezza. P.es., si consideri lo scompattare un pacchetto o
lo scaricare file da github: un file .vimrc ivi contenuto
potrebbe essere un "cavallo di Troia".
MEGLIO NON IMPOSTARE QUEST'OPZIONE!
In alternativa, si può definire un autocomando nel vostro .vimrc
che imposti opzioni se si sta lavorando su una data directory.

Se attivate quest'opzione è consigliabile impostare l'opzione
'secure' option (vedere |initialization|). L'uso di un .exrc, .vimrc
o .gvimrc locale è un potenziale rischio per la sicurezza, usare con
cautela! Vedere anche |.vimrc| e |gui-init|.
Quest'opzione non può essere impostata da |modeline|, o nel |sandbox|,
per motivi di sicurezza.

'fileencoding' *'fenc'* *E213*
'fileencoding' 'fenc' stringa (default: "")
locale al buffer
{disponibile solo se compilato con la funzionalità
|+multi_byte|}
{non in Vi}
Imposta la codifica dei caratteri per il file in questo buffer.

Quando 'fileencoding' è diverso da 'encoding', la conversione avrà
luogo al momento di leggere e scrivere il file. Per la lettura,
vedere più sotto.
Quando 'fileencoding' non è impostato, lo stesso valore di 'encoding'
sarà usato (nessuna conversione quando si legge o scrive il file).
Nessun messaggio di errore sarà emesso quando il valore è impostato,
ma solo quando sarà usato, solo nella scrittura di un file.
Una conversione sarà pure effettuata quando 'encoding' e 'fileencoding'
sono entrambi codifiche Unicode e 'fileencoding' non è utf-8. Questo
si deve al fatto che all'interno di Vim Unicode è sempre gestito con
la codifica utf-8.
ATTENZIONE: La Conversione può generare perdita di
informazione! Quando 'encoding' vale "utf-8" o qualche
altra codifica Unicode, la conversione è generalmente
effettuata in modo che la conversione opposta
produca lo stesso testo. Quando 'encoding' non vale "utf-8"
alcuni caratteri possono andare persi!

Vedere 'encoding' per i possibili valori. Inoltre, possono essere
specificati valori che siano gestiti dal convertitore, vedere
|mbyte-conversion|.

Quando si legge un file, 'fileencoding' sarà impostato come
'fileencodings'. Per leggere un file con una codifica a vostra scelta
non si deve impostate 'fileencoding', va usato invece l'argomento

|**++enc**|. C'è un'eccezione: quando 'fileencodings' è nullo, si usa il valore di 'fileencoding'.
Per un file nuovo si usa il valore globale di 'fileencoding'.

I prefissi "8bit-" e "2byte-" non hanno senso qui, e sono ignorati. Quando l'opzione è impostata, il valore è convertito in minuscolo. Quindi potete impostarla anche usando lettere maiuscole. Caratteri '_' sono rimpiazzati con '-'. Se un nome è riconosciuto come incluso nella lista di 'encoding', sarà rimpiazzato dal nome standard. Ad esempio "ISO8859-2" diviene "iso-8859-2".

Quando quest'opzione è impostata, all'inizio della modifica di un file, l'opzione 'modified' è impostata, perché il file sarà differente nel momento in cui verrà riscritto.

Tenete presente che il cambiamento di 'fenc' da una modeline avviene DOPO che il testo è stato letto, e quindi avrà effetto quando il file sarà riscritto. Se impostate 'fenc' in una modeline, potreste impostare anche 'nomodified', per evitare di no riuscire ad eseguire il comando ":q".

Quest'opzione non si può cambiare quando 'modifiable' è inibito.

'fe'
NOTA: Prima della versione 6.0 quest'opzione specificava la codifica per Vim a livello generale, il che era errato. Adesso usate invece 'encoding' . Il vecchio nome abbreviato dell'opzione era 'fe', che ora non si usa più.

'fileencodings' *'fencs'*
'fileencodings' 'fencs' stringa (default: "ucs-bom",
"ucs-bom,utf-8,default,latin1" quando
'encoding' è impostato a un valore Unicode)
globale
{disponibile solo se compilato con la funzionalità
|**+multi_byte**|}
{non in Vi}
Questa è una lista di codifiche carattere considerate quando si inizia a modificare un file già esistente. Quando si legge un file, Vim tenta di usare la prima codifica caratteri della lista. Se si riscontra un errore, si passa alla successiva codifica presente nella lista. Quando si trova una codifica che non dà problemi, 'fileencoding' è impostato con la codifica stessa. Se nessuna codifica funziona, 'fileencoding' è impostato ad una stringa nulla, il che causa l'utilizzo del valore di 'encoding' .
ATTENZIONE: La Conversione può generare perdita di informazione! Quando 'encoding' vale "utf-8" la conversione è generalmente effettuata in modo che la conversione opposta produce lo stesso testo. Quando 'encoding' non vale "utf-8" alcuni caratteri non-ASCII possono andare persi! Si può usare l'argomento |**++bad**| per specificare cosa fare con i caratteri che non è possibile convertire.
Per un file vuoto o che contenga solo caratteri ASCII la maggior parte delle codifiche funziona, e sarà usato il primo elemento della lista di 'fileencodings' (tranne se è "ucs-bom", che richiede la presenza del BOM). Se preferite un'altra codifica, usate un autocomando di evento BufReadPost per controllare che sia usata la vostra codifica preferita. Ad es.: >
```
au BufReadPost * if search('\S', 'w') == 0 |
\ set fenc=iso-2022-jp | endif
```
< In questo modo si imposta 'fileencoding' a "iso-2022-jp" se il file non contiene caratteri diversi dallo spazio.
Se si usa l'argomento |**++enc**| il valore di 'fileencodings' non è usato.
Nota 'fileencodings' non è usato per un file nuovo, nel qual caso si usa invece il valore globale di 'fileencoding'. Potete impostarlo con: >
```
:setglobal fen=iso-8859-2
```
< Ciò significa che a un file non esistente può essere attribuita una codifica differente da quella per un file vuoto.
Il valore speciale "ucs-bom" si può usare per controllare la presenza di un BOM (Byte Order Mark) Unicode all'inizio del file. Non deve essere preceduto da "utf-8" o da un'altra codifica Unicode per essere

utilizzato correttamente. Una indicazione di una codifica a 8 bit (ad es., "latin1") dovrebbe venire per ultima, perché Vim non è in grado di accertare un errore, e quindi la codifica è sempre accettata. Il valore speciale "default" può essere usato per la codifica presa dalle variabili d'ambiente. Questo è il valore di default per 'encoding'. È utile quando 'encoding' è impostato a "utf-8" e il vostro ambiente usa una codifica non-latin1, come il Russo. Quando 'encoding' è "utf-8" e il file contiene una sequenza di byte non valida, non verrà riconosciuto come utf-8. Si può usare il comando |**8g8**| per trovare la sequenza di byte non valida.

VALORI ERRATI:	COSA C'È DI SBAGLIATO:
latin1,utf-8	verrà sempre usato "latin1"
utf-8,ucs-bom,latin1	BOM non riconosciuto in un file utf-8
cp1250,latin1	verrà sempre usato "cp1250"

Se 'fileencodings' non è impostato, 'fileencoding' non sarà cambiato. Vedere 'fileencoding' per i possibili valori. L'impostazione di quest'opzione produrrà effetti solo alla successiva lettura di un file.

'fileformat' *'ff'*

'fileformat' 'ff' stringa (default MS-DOS, MS-Windows, OS/2: "dos",
default Unix: "unix",
default Macintosh: "mac")
locale al buffer
{non in Vi}

Quest'opzione definisce la stringa <EOL> (End of Line, fine della riga) per il buffer corrente, e che viene usata per leggere/scrivere il buffer da/in un file:

dos	<CR> <NL>
unix	<NL>
mac	<CR>

Quando si usa "dos", un eventuale CTRL-Z a fine file è ignorato. Vedere |**file-formats**| e |**file-read**|. Per la codifica caratteri del file vedere 'fileencoding'. Quando l'opzione 'binary' è impostata, il valore di 'fileformat' è ignorato, la lettura/scrittura dei file funziona come se fosse impostato a "unix". Quest'opzione è impostata automaticamente quando si inizia a modificare un file, se 'fileformats' è specificato e 'binary' è a off. Quando quest'opzione è impostata dopo aver iniziato una sessione di modifica di un file, il flag 'modified' viene attivata, perché il file in questione sarà differente quando verrà riscritto. Quest'opzione non si può cambiare se 'modifiable' è a off [ossia quando il file è designato come non modificabile]. Per compatibilità all'indietro: Quando quest'opzione è impostata a "dos", 'textmode' viene impostato, altrimenti 'textmode' è a off.

'fileformats' *'ffs'*

'fileformats' 'ffs' stringa (default:
Vim+Vi MS-DOS, MS-Windows OS/2: "dos,unix",
Vim Unix: "unix,dos",
Vim Mac: "mac,unix,dos",
Vi Cygwin: "unix,dos",
Vi altri: "")
globale
{non in Vi}

Questa è una lista di formati di fine riga (<EOL>), da provare nell'ordine, quando si inizia a modificare un nuovo buffer e quando si legge un file in un buffer già esistente:

- Se non specificata, verrà sempre usato il formato definito con 'fileformat'. Il valore non sarà impostato automaticamente.
- Quando impostato con un solo nome, il formato specificato sarà sempre usato quando si apre un nuovo buffer. 'fileformat' è impostato in conseguenza per quel buffer. Il nome specificato in 'fileformats' sarà usato quando un file è letto in un buffer esistente, a prescindere dal valore impostato di 'fileformat' per quel buffer.
- Quando è presente più di un nome, separati da virgola, sarà fatta una determinazione automatica di <EOL> al momento della lettura di un file. Quando si comincia a modificare un file, viene fatto un controllo per <EOL>:
  1. Se tutte le righe finiscono con <CR><NL>, [<Ritorno_Carrello>, <A_Capo>] e 'fileformats' contiene "dos", 'fileformat' è

impostato a "dos".
2. Se si trova un <NL> e 'fileformats' contiene "unix", 'fileformat' è impostato a "unix". Nota Quando si trova un <NL> non preceduto da <CR>, viene preferito "unix" rispetto a "dos".
3. Se 'fileformat' non è stato ancora impostato e se viene trovato un <CR>, e se 'fileformats' comprende "mac", 'fileformat' viene impostato a "mac".
"unix" non è presente, oppure non viene trovato alcun <NL> nel file, e
"dos" non è presente, oppure non esiste alcuna sequenza <CR><NL> nel file.
Tranne nel caso in cui: era stato scelto "unix", ma c'è un <CR> prima del primo <NL> e risultano esserci più <CR> che <NL> nelle prime righe del file, allora 'fileformat' è impostato a "mac".
4. Se 'fileformat' non è ancora impostato, si usa il primo nome presente nella lista 'fileformats'.

Quando un file viene letto in un buffer già esistente, si segue la stessa procedura, ma l'effetto è come se 'fileformat' venga impostato esclusivamente per il file che viene letto, mentre il valore dell'opzione non viene cambiato.
Quando 'binary' è impostato, il valore di 'fileformats' non è utilizzato.

Quando Vim viene invocato con un buffer vuoto, il primo elemento nella lista è quello utilizzato. È possibile modificare la scelta di default specificando 'fileformat' in un file .vimrc.

Per sistemi con una "fine-riga" <EOL> simile al Dos (<CR><NL>), quando si leggono file tramite il comando ":source" o file vimrc, può venire effettuata una determinazione automatica dell'<EOL>:
- Se 'fileformats' non è impostato, non si fa una determinazione automatica. Verrà usato il formato Dos.
- Se 'fileformats' è impostato e contiene uno o più nomi, si fa una determinazione automatica. Questa è basata sul primo <NL> contenuto nel file: Se è preceduto da un <CR>, si usa il formato Dos, altrimenti si usa il formato Unix.

Vedere anche |**file-formats**|.
Per compatibilità all'indietro: Quando quest'opzione è impostata come una stringa nulla oppure con un solo formato (senza alcuna virgola), l'opzione 'textauto' è messa a off, altrimenti 'textauto' è impostata.
NOTA: Quest'opzione è impostata al valore di default di Vi impostando 'compatible' e al valore di default di Vim quando 'compatible' viene messa a off.

'fileignorecase' *'fic'* *'nofileignorecase'* *'nofic'*
'fileignorecase' 'fic' booleana (per default è impostata per sistemi in cui lettere maiuscole/minuscole sono normalmente equivalenti nei nomi di file)
globale
{non in Vi}

Quando è impostata a on, le lettere maiuscole/minuscole sono considerate equivalenti quando compaiono in nomi di file o di directory.
Vedere 'wildignorecase' per ignorare maiuscole/minuscole solo quando si stia effettuando un completamento.

'filetype' *'ft'*
'filetype' 'ft' stringa (default: "")
locale al buffer
{non in Vi}

Quando quest'opzione è impostata, l'autocomando per l'evento FileType viene attivato.
Tutti gli autocomandi corrispondenti al valore di quest'opzione saranno eseguiti. Quindi il valore di 'filetype' è usato al posto del nome del file.
Altrimenti quest'opzione non sempre rispecchia il tipo del file corrente.
Quest'opzione è solitamente impostata quando viene determinato il tipo di file. Per abilitarla, usate il comando ":filetype on".
|**:filetype**|
Impostare quest'opzione ad un valore diverso è molto utile in una modeline, per un file il cui tipo file non viene riconosciuto automaticamente.

Ad esempio, per un file IDL:
```
        /* vim: set filetype=idl : */ ~
```
|**FileType**| |**filetypes**|
Quando un punto (.) è presente nel valore, indica una separazione tra due tipi di file. Esempio:
```
        /* vim: set filetype=c.doxygen : */ ~
```
In questo caso si usa dapprima il tipo file "c", e poi il "doxygen". Ciò si applica sia a filetype plugin che a file di sintassi. Può essere presente più di un punto.
Quest'opzione non è copiata in un altro buffer, a prescindere dai flag 's' o 'S' in 'cpoptions'.
Solo caratteri normali possono essere usati nei nomi di file, i caratteri "/*?[|<>" non sono validi.

'fillchars' *'fcs'*
'fillchars' 'fcs' stringa (default: "vert:|,fold:-")
globale
{non in Vi}
{non disponibile se compilato senza le funzionalità |**+windows**| e |**+folding**|}
Caratteri per riempire la riga di status e i separatori verticali. È una lista di elementi separati dalla virgola:

elemento	default	utilizzato per ~
stl:c	' ' o '^'	riga di status della finestra corrente
stlnc:c	' ' o '='	riga di status della finestra non-corrente
vert:c	'|'	separatori verticali |**:vsplit**|
fold:c	'-'	riempitivo per 'foldtext'
diff:c	'-'	righe cancellate nell'opzione 'diff'

Per valori non specificati, viene usato il default. Per "stl" e "stlnc" si usa lo spazio se c'è evidenziazione, altrimenti '^' o '='.

Esempio: >
```
    :set fillchars=stl:^,stlnc:=,vert:\|,fold:-,diff:-
```
< Simile al default, ma questi caratteri verranno usati anche nel caso in cui ci sia evidenziazione.

Per "stl" e "stlnc" sono supportati solo valori costituiti da un unico byte.

L'evidenziazione è usata per questi elementi:

elemento	gruppo evidenziazione ~	
stl:c	StatusLine	|**hl-StatusLine**|
stlnc:c	StatusLineNC	|**hl-StatusLineNC**|
vert:c	VertSplit	|**hl-VertSplit**|
fold:c	Folded	|**hl-Folded**|
diff:c	DiffDelete	|**hl-DiffDelete**|

'fixendofline' *'fixeol'* *'nofixendofline'* *'nofixeol'*
'fixendofline' 'fixeol' booleana (default: on)
locale al buffer
{non in Vi}
Quando si scrive un fine e quest'opzione è a on, <EOL> a fine fine verrà aggiunto, se non presente. Si imposti a off quest'opzione se si desidera mantenere la situazione presente nel file originale.
Quando l'opzione 'binary' è a on, il valore di quest'opzione non è utilizzato.
Vedere l'opzione 'endofline'.

'fkmap' *'fk'* *'nofkmap'* *'nofk'*
'fkmap' 'fk' booleana (default: off) *E198*
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+rightleft**|}
Se impostata, la tastiera è mappata per i caratteri Farsi [Persiani]. Normalmente si imposta 'allowrevins' e si usa CTRL-_ in modo Insert per attivare/disattivare quest'opzione |**i_CTRL-_**|. Vedere |**farsi**|.

'foldclose' *'fcl'*
'foldclose' 'fcl' stringa (default: "")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+folding**|}
Se impostata a "all", una piegatura è chiusa se il cursore non si trova all'interno di essa e se il suo livello è superiore a 'foldlevel'. Utile se volete che le piegature si chiudano da sole quando vi spostate fuori da esse.

'foldcolumn' *'fdc'*
'foldcolumn' 'fdc' numero (default: 0)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+folding**|}
Se diversa da zero, una colonna della larghezza specificata viene visualizzata a fianco della finestra, ed indica le piegature aperte e chiuse. Il valore massimo è 12.
Vedere |**folding**|.

'foldenable' *'fen'* *'nofoldenable'* *'nofen'*
'foldenable' 'fen' booleana (default: on)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+folding**|}
Se inattiva, tutte le piegature sono aperte. Quest'opzione si può usare per alternare rapidamente fra la visione di tutto il testo senza piegature e la visione del testo con piegature (include le piegature aperte o chiuse manualmente). Può essere attivato o inattivato con il comando |**zi**|. Le colonne 'foldcolumn' resteranno bianche quando 'foldenable' è inattivo.
Quest'opzione è impostata da comandi che creano una nuova piegatura o che chiudono una piegatura. Vedere |**folding**|.

'foldexpr' *'fde'*
'foldexpr' 'fde' stringa (default: "0")
locale alla finestra
{non in Vi}
{non disponibile se compilato senza le funzionalità
|**+folding**| o |**+eval**|}
È l'espressione usata quando 'foldmethod' è "expr". Viene valorizzata ad ogni riga, per ottenere il livello di piegatura della riga. Vedere |**fold-expr**|.

L'espressione sarà valutata nel |**sandbox**|, se è stata impostata da una modeline; vedere |**sandbox-option**|.
Quest'opzione non può essere impostata da |**modeline**| se l'opzione 'diff' è attiva.

Non è consentito cambiare testi o passare a un'altra finestra mentre è in corso la valutazione di 'foldexpr' |**textlock**|.

'foldignore' *'fdi'*
'foldignore' 'fdi' stringa (default: "#")
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+folding**|}
Usato solo quando 'foldmethod' è "indent". Righe che iniziano con caratteri specificati in 'foldignore' avranno il livello di piegatura delle righe adiacenti. Eventuali spazi bianchi iniziali sono ignorati, nel controllare questo carattere. Il valore di default "#" va bene per programmi scritti in C. Vedere |**fold-indent**|.

'foldlevel' *'fdl'*
'foldlevel' 'fdl' numero (default: 0)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+folding**|}

Imposta il livello di piegatura: piegature con un livello più alto di
questo saranno chiuse.
Impostando quest'opzione a zero tutte le piegature saranno chiuse.
Numeri più alti chiuderanno un numero minore di piegature.
Quest'opzione è impostata da comandi come |**zm**|, |**zM**| e |**zR**|.
Vedere |**fold-foldlevel**|.

'foldlevelstart' *'fdls'*
'foldlevelstart' 'fdls' numero (default: -1)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+folding**|}
Imposta 'foldlevel' quando si inizia ad editare un altro buffer in una
finestra.
Utile per iniziare sempre ad editare con tutte le piegature chiuse
(valore zero), alcune piegature chiuse (uno) o nessuna piegatura
chiusa (99).
Questo si fa prima di leggere qualsiasi modeline, e per questo una
eventuale impostazione in una modeline prevale su quest'opzione.
Anche se si inizia ad editare un file in |**diff-mode**| quest'opzione
viene ignorata, e tutte le piegature vengono chiuse.
Questo si fa anche prima di eseguire gli autocomandi BufReadPre per
permettere ad un autocomando di decidere il valore di 'foldlevel'
per determinati file.
Se il valore è negativo, non viene utilizzato.

'foldmarker' *'fmr'* *E536*
'foldmarker' 'fmr' stringa (default: "{{{,}}}")
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+folding**|}
Il delimitatore di inizio e fine usato quando 'foldmethod' è "marker".
Ci deve essere una virgola, come separatore fra la stringa di inizio e
quella di fine. Il delimitatore deve essere una semplice stringa di
caratteri (una espressione regolare rallenterebbe troppo
l'esecuzione).
Vedere |**fold-marker**|.

'foldmethod' *'fdm'*
'foldmethod' 'fdm' stringa (default: "manual")
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+folding**|}
Il tipo di piegatura usato per la finestra corrente. Valori possibili:

|**fold-manual**|	manual	Le piegature sono create manualmente.
|**fold-indent**|	indent	Righe con la stessa indentatura appartengono alla stessa piegatura.
|**fold-expr**|	expr	'foldexpr' stabilisce il livello di piegatura di una riga.
|**fold-marker**|	marker	Delimitatori (mark) sono usati per specificare la piegatura.
|**fold-syntax**|	syntax	Elementi di evidenziazione sintattica sono usati per specificare la piegatura.
|**fold-diff**|	diff	Piegatura costituita da testo che non è cambiato.

'foldminlines' *'fml'*
'foldminlines' 'fml' numero (default: 1)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+folding**|}
Imposta il numero minimo di righe sullo schermo sopra il quale una
piegatura è visualizzata come chiusa. Vale anche per piegature chiuse
col metodo manuale. Con il valore di default di uno, una piegatura
può essere chiusa solo se comprende due o più righe dello schermo.
Impostare a zero consente di chiudere piegature che consistono solo di
una linea del file.
Nota Questo ha effetto solo su quel che viene visualizzato. Dopo aver
usato "zc" per chiudere una piegatura, che è visualizzata aperta

perché più piccola di 'foldminlines', un ulteriore "zc" può chiudere una piegatura che la contenga.

'foldnestmax' *'fdn'*
'foldnestmax' 'fdn' numero (default: 20)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità |+folding|}

Imposta la massima indentatura per i metodi di piegatura "indent" e "syntax". Questo evita di creare un numero eccessivo di piegature. L'uso di un numero maggiore di 20 non funziona, perché il limite interno è impostato a 20.

'foldopen' *'fdo'*
'foldopen' 'fdo' stringa (default: "block,hor,mark,percent,quickfix, search,tag,undo")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |+folding|}

Specifica per quali tipi di comando una piegatura verrà aperta, quando il comando posizionerebbe il cursore in una piegatura chiusa. È una lista di elementi, separati da virgola.
NOTA: Quando il comando fa parte di una mappatura, quest'opzione non è utilizzata. Aggiungere il comando |zv| alla mappatura per ottenere lo stesso effetto.
(motivo: la mappatura stessa potrebbe voler controllare l'apertura delle piegature)

elemento	comandi ~
all	ogni comando
block	"(", "{", "[[", "[{", etc.
hor	movimenti orizzontali: "l", "w", "fx", etc.
insert	ogni comando in modo Insert
jump	salti "lunghi": "G", "gg", etc.
mark	salto ad un mark: "'m", CTRL-O, etc.
percent	"%"
quickfix	":cn", ":crew", ":make", etc.
search	ricerca di un'espressione: "/", "n", "*", "gd", etc. (non per un'espressione di ricerca in un comando ":") Anche per |[s| e |]s|.
tag	salto ad un tag: ":ta", CTRL-T, etc.
undo	undo o redo: "u" e CTRL-R

Quando un comando di movimento è usato per un operatore (ad es., "dl" o "y%") quest'opzione non è utilizzata. Ovvero l'operatore comprenderà l'intera piegatura chiusa.
Nota I movimenti verticali non sono presenti, perché ciò renderebbe molto difficile "passare sopra" ad una piegatura chiusa.
In modo Insert la piegatura contenente il cursore sarà sempre aperta mentre si inserisce del testo.
Per chiudere piegature si può ri-applicare 'foldlevel' col comando |zx| o impostare l'opzione 'foldclose' a "all".

'foldtext' *'fdt'*
'foldtext' 'fdt' stringa (default: "foldtext()")
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità |+folding|}

Un'espressione usata per specificare il testo da visualizzare dove si trova una piegatura chiusa. Vedere |fold-foldtext|.

L'espressione sarà valutata nel |sandbox|, se è stata impostata da una modeline; vedere |sandbox-option|.

Non è consentito cambiare testi o passare a un'altra finestra mentre è in corso la valutazione di 'foldtext' |textlock|.

'formatexpr' *'fex'*
'formatexpr' 'fex' stringa (default: "")

locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+eval**|}
Espressione che viene valutata per formattare un gruppo di righe
da parte dell'operatore |**gq**| o per la formattazione automatica (vedere
'formatoptions'. Quando quest'opzione è una stringa nulla, si usa
'formatprg'.

La variabile |**v:lnum**| contiene il numero della prima riga da
formattare.
La variabile |**v:count**| contiene il numero di righe da formattare.
La variabile |**v:char**| contiene il carattere che sarà inserito se
si sta valutando l'espressione durante una
formattazione automatica. Quest'ultimo può essere
nullo. Aspettate a inserirlo!

Esempio: >
:set formatexpr=mylang#Format()
< Questo invoca la funzione mylang#Format() nel file
autoload/mylang.vim contenuto in 'runtimepath'. |**autoload**|

L'espressione viene anche valutata quando si imposta 'textwidth'
e si aggiunge del testo che supera quel limite.
Ciò succede nelle stesse condizioni in cui si usa
la formattazione interna. In questo caso, assicuratevi che il cursore
sia mantenuto nello stesso punto, relativamente al testo!
La funzione |**mode()**| restituirà in questa situazione "i" o "R".

Quando la funzione restituisce un valore diverso da zero, Vim tornerà
a usare il meccanismo di formattazione interno.

L'espressione sarà valutata nel |**sandbox**|, se è stata impostata da una
modeline; vedere |**sandbox-option**|. Ciò fa sì che l'opzione non
funzioni, in quanto la modifica del testo contenuto nel buffer non è
ammessa.
NOTA: Quest'opzione è impostata a "" se si attiva 'compatible'.

'formatoptions' *'fo'*
'formatoptions' 'fo' stringa (default Vim: "tcq", default Vi: "vt")
locale al buffer
{non in Vi}
Questa è una sequenza di lettere che indica come effettuare la
formattazione. Vedere |**fo-table**|. Quando l'opzione 'paste' è
impostata non viene effettuata alcuna formattazione (come succede se
'formatoptions' è una stringa nulla). Si possono inserire virgole,
per migliorare la leggibilità.
Per evitare problemi con flag che potrebbero essere aggiunti in
futuro, usare la notazione "+=" e "-=" di ":set" |**add-option-flags**|.
NOTA: Quest'opzione è impostata al valore di default di Vi
impostando 'compatible' e al valore di default di Vim quando
'compatible' viene messa a off.

'formatlistpat' *'flp'*
'formatlistpat' 'flp' stringa (default: "^\s*\d\+[\]:.)}\t]\s*")
locale al buffer
{non in Vi}
Un'espressione regolare usata per riconoscere una testata di lista.
Serve per il flag "n" in 'formatoptions'.
L'espressione regolare deve corrispondere esattamente al testo che
detta l'indentatura per la riga sotto di esso. Si può usare |**/\ze**|
per segnare la fine della corrispondenza, mentre è in corso il
controllo di caratteri ulteriori. Ci deve essere almeno un
carattere, dopo l'espressione regolare. Se viene trovata una
corrispondenza con una riga intera, è considerata come se non ci
fosse alcuna corrispondenza.
Il default riconosce un numero, seguito da un segno di interpunzione
facoltativo e da uno spazio bianco.

'formatprg' *'fp'*
'formatprg' 'fp' stringa (default: "")
globale o locale al buffer |**global-local**|
{non in Vi}

Programma esterno da usare per formattare le righe selezionate con l'operatore |**gq**|. Il programma deve prendere in input "stdin" e inviare l'output a "stdout". Il comando Unix "fmt" è uno di questi programmi.
Se l'opzione 'formatexpr' non è una stringa nulla, essa verrà usata. Altrimenti, se l'opzione 'formatprg' è una stringa nulla, sarà usata la funzione interna di formattazione |**C-indenting**|.
Le variabili d'ambiente vengono valutate |**:set_env**|. Vedere |**option-backslash**| per inserire spazi e backslash.
L'espressione può essere valutata nel |**sandbox**|, vedere |**sandbox-option**|.
Quest'opzione non può essere impostata da una |**modeline**| o nel |**sandbox**|, per motivi di sicurezza.

'fsync' *'fs'* *'nofsync'* *'nofs'*
'fsync' 'fs' booleana (default: on)
globale
{non in Vi}

Quando attivata, la funzione di sistema fsync() sarà invocata dopo aver scritto un file. Ciò provocherà la riscrittura fisica del file su disco, in modo da essere certi che sia stato scritto anche in File System che fanno del journaling solo per i metadati (ossia per i dati relativi ai dati veri). Questo costringerà l'hard disk a muoversi parecchio, su sistemi Linux che vanno in modo Laptop, e ciò può essere indesiderabile in alcuni casi. Restate comunque avvisati che inibire quest'opzione aumenta le possibilità di perdite di dati dopo una caduta di sistema. In sistemi che non hanno un'implementazione di fsync(), questa variabile è sempre a off.
Vedere anche 'swapsync' per controllare fsync() verso i file di swap.
'fsync' vale anche per |**writefile()**|, a meno che si usi un flag per inibire questa funzionalità.

'gdefault' *'gd'* *'nogdefault'* *'nogd'*
'gdefault' 'gd' booleana (default: off)
globale
{non in Vi}

Se impostata, il flag 'g' di ":substitute" è attiva per default. Ciò vuol dire che ogni corrispondenza trovata in una riga verrà modificata, e non solo la prima che viene trovata. Quando un flag 'g' viene inserito in un comando ":substitute", questo alternerà fra la sostituzione di una o di tutte le corrispondenze. Vedere |**complex-change**|.

comando	'gdefault' on	'gdefault' off ~
:s///	sostit. tutte	sostit. prima
:s///g	sostit. prima	sostit. tutte
:s///gg	sostit. tutte	sostit. prima

NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.
DEPRECATO: L'impostazione di quest'opzione può determinare errori in plugin che non tengono conto dell'opzione stessa, Inoltre, molti utenti potrebbero essere confusi per il fatto che aggiungendo il flag /g si verificano effetti opposti a quelli che normalmente produce.

'grepformat' *'gfm'*
'grepformat' 'gfm' stringa (default "%f:%l:%m,%f:%l%m,%f %l%m")
globale
{non in Vi}

Descrizione del formato delle righe di output del comando "grep". È questa una stringa in formato simile a quello della funzione [C] scanf, ed usa lo stesso formato dell'opzione 'errorformat': vedere |**errorformat**|.

'grepprg' *'gp'*
'grepprg' 'gp' stringa (default: "grep -n ",
Unix: "grep -n $* /dev/null",
Win32: "findstr /n" o "grep -n",
VMS: "SEARCH/NUMBERS ")
globale o locale al buffer |**global-local**|
{non in Vi}

Programma da usare per il comando ":grep". Quest'opzione può contenere caratteri '%' e '#', che sono valutati come quando sono usati in una riga-comandi.

Il segnaposto "$*" consente, volendo, di specificare dove vanno
inseriti gli argomenti.
Le variabili d'ambiente vengono valutate |**:set_env**|.
Vedere |**option-backslash**| per inserire spazi e backslash.
Quando il vostro comando "grep" accetta l'argomento "-H", usate
quanto segue per far sì che ":grep" funzioni correttamente anche con
un solo file: >
```
:set grepprg=grep\ -nH
```
< Di valore speciale: Quando 'grepprg' è impostato a "internal" il
comando |**:grep**| funziona come |**:vimgrep**|, |**:lgrep**| come |**:lvimgrep**|,
|**:grepadd**| come |**:vimgrepadd**| e |**:lgrepadd**| come |**:lvimgrepadd**|.
Vedere anche la sezione |**:make_makeprg**|, dato che molti dei commenti
ivi contenuti sono validi anche per 'grepprg'.
Per Win32, il default è "findstr /n" se "findstr.exe" è disponibile,
altrimenti è "grep -n".
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|,
per motivi di sicurezza.

'guicursor' *'gcr'* *E545* *E546* *E548* *E549*
'guicursor' 'gcr' stringa (default: "n-v-c:block-Cursor/lCursor,
ve:ver35-Cursor,
o:hor50-Cursor,
i-ci:ver25-Cursor/lCursor,
r-cr:hor20-Cursor/lCursor,
sm:block-Cursor
-blinkwait175-blinkoff150-blinkon175",
per MS-DOS e Win32 console:
"n-v-c:block,o:hor50,i-ci:hor15,
r-cr:hor30,sm:block")
globale
{non in Vi}
{disponibile solo se compilato con la GUI abilitata,
e per la console MS-DOS e Win32}
Quest'opzione dice a Vim come dovrebbe essere il cursore in
differenti modalità. Funziona completamente solo nella GUI. In una
console MSDOS o Win32, solo l'altezza del cursore può essere cambiata.
Questo si può fare richiedendo un cursore rettangolare, oppure
specificando una percentuale, per un cursore verticale od orizzontale.
Per una console si usano le sequenze protette di caratteri 't_SI',
't_SR' e 't_EI'.

L'opzione è una lista di parti, separate da virgola. Ogni parte
consiste di una mode-list e di una argument-list:
mode-list:argument-list,mode-list:argument-list,..
La mode-list è una lista, separata da ":" di questi modi:

n	modo Normal
v	modo Visual
ve	modo Visual con 'selection' "exclusive" (come 'v', se l'opzione non è specificata)
o	modo Operator-pending
i	modo Insert
r	modo Replace
c	modo Command-line Normal (append)
ci	modo Command-line Insert
cr	modo Command-line Replace
sm	showmatch in modo Insert
a	tutti i modi

L'argument-list è un elenco separato da trattini ("-") dei seguenti
argomenti:

hor{N}	barra orizzontale, {N} percentuale dell'altezza del carattere
ver{N}	barra verticale, {N} percentuale della larghezza del carattere
block	cursore rettangolare, riempie lo spazio intero del carattere

[sono uno dei tre qui sopra dovrebbe essere presente]
blinkwait{N} *cursor-blinking*
blinkon{N}
blinkoff{N}
tempi di lampeggiamento per il cursore: blinkwait è il
tempo che deve passare prima che il cursore cominci a
lampeggiare, blinkon è il tempo in cui il cursore è
visibile, e blinkoff è il tempo in cui il cursore non

è visualizzato. I tempi sono espressi in
millisecondi. Quando uno dei numeri è a zero, non c'è
lampeggiamento. Il default è:
"blinkwait700-blinkon400-blinkoff250".
Questi numeri vengono usati in assenza di indicazioni
esplicite. Ciò significa che il lampeggiamento è
abilitato per default. Per inibire il lampeggiamento,
potete usare "blinkon0". Il cursore lampeggia solo
quando Vim è in attesa di ricevere dell'input, non
mentre un comando è in esecuzione.
Per far lampeggiare il cursore in un xterm, vedere
|**xterm-blink**|.
{group-name}
il nome di un gruppo di evidenziazione, che imposta il
colore e il font per il cursore.
{group-name}/{group-name}
Due nomi di gruppi di evidenziazione, il primo dei
quali viene usato quando non ci sono mappature di
linguaggio, l'altro se ce ne sono. Vedere
|**language-mapping**|.

Esempi di parti:
n-c-v:block-nCursor in modo Normal, Command-line e Visual, usare
un cursore rettangolare con colori specificati
dal gruppo di evidenziazione "nCursor"
i-ci:ver30-iCursor-blinkwait300-blinkon200-blinkoff150
In modo Insert e Command-line Insert, usare
un cursore verticale al 30% con colori
specificati dal gruppo di evidenziazione
"iCursor". Lampeggiare un po' più
velocemente.

Il modo 'a' è differente. Imposterà l'argument-list data per tutti i
modi. Non utilizza per nulla i default. Questo si può usare per
dare una impostazione comune a tutti i modi. Ad esempio, per inibire
il lampeggiamento: "a:blinkon0"

Esempi di evidenziazione del cursore: >
:highlight Cursor gui=reverse guifg=NONE guibg=NONE
:highlight Cursor gui=NONE guifg=bg guibg=fg
<

'guifont' *'gfn'*
E235 *E596*
'guifont' 'gfn' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la GUI abilitata}
Questa è una lista di font da usare per la versione GUI di Vim.
Nella sua forma più semplice il valore è solo un nome di font. Se il
font non è disponibile, verrà emesso un messaggio di errore. Per
cercare altri font, si può specificare una lista, con nomi di font
separati da virgola. Il primo font trovato disponibile sarà usato.

Nei sistemi in cui 'guifontset' è supportato (X11) e 'guifontset' non
è nullo, 'guifont' non è utilizzato.

Nota: Per quel che riguarda le GUI GTK, non viene emesso nessuno
messaggio di errore a fronte di nomi invalidi, e il primo elemento
trovato nella lista è sempre scelto e utilizzato.
Ciò succede perché, invece di indentificare un dato nome con un
carattere, le GUI GTK lo usano per costruire un'espressione regolare
e con quella tentare di trovare il carattere che meglio corrisponda
all'espressione regolare, tra i caratteri disponibili, e in questo
modo la ricerca ha sempre successo. Specificare nome non valido non
ha importanza, perché una serie di proprietà di carattere diverse
dal nome permetteranno di identificare comunque un carattere.

Gli spazi dopo una virgola sono ignorati. Per includere una virgola
in un nome di font, fatela precedere da un backslash [\]. Vedere
anche
|**option-backslash**|. Ad esempio: >
:set guifont=Screen15,\ 7x13,font\\,con\\,virgole
< richiede a Vim di provare ad usare il font "Screen15" per primo, e se

questo non è disponibile, di tentare di usare "7x13" e poi, se serve, "font,con,virgole".

Se non è possibile caricare nessuno dei font specificati, Vim manterrà le impostazioni correnti. Se viene specificata una lista di font vuota, Vim tenterà di usare altre impostazioni di risorse (per X, userà la risorsa Vim.font), e infine tenterà alcuni default interni ("7x13" nel caso di X). I nome di font specificati dovrebbero essere forn in formato "normale". Vim cercherà i relativi font in grassetto e in corsivo.

Per Win32, GTK, Motif, Mac OS e Photon: >
```
:set guifont=*
```
< invocherà un dialogo di richiesta font, scegliete quello che preferite.

I nome di font dipendono dall GUI utilizzata.
Vedere |**setting-guifont**| per indicazioni su come impostare 'guifont' per vari sistemi.

Per le GUI GTK+ 2 e 3 i nomi di font sono di questo tipo: >
```
:set guifont=Andale\ Mono\ 11
```
< Questo è tutto. Gli XLFD non sono più accettati. Per la lingua cinese risulta che quel che segue funzioni bene: >
```
if has("gui_gtk2")
  set guifont=Bitstream\ Vera\ Sans\ Mono\ 12,Fixed\ 12
  set guifontwide=Microsoft\ Yahei\ 12,WenQuanYi\ Zen\ Hei\ 12
endif
```
<

Per Mac OSX potete usare qualcosa del tipo: >
```
:set guifont=Monaco:h10
```
< Vedere anche 'macatsui', che può essere utile per risolvere problemi di visualizzazione dello schermo.

E236

Nota I font devono essere mono-spaziati (tutti i caratteri devono avere la stessa larghezza). Fa eccezione GTK: qualsiasi font è accettato, ma i font mono-spaziati hanno un aspetto migliore.

Per vedere in anteprima un font in X11, potreste usare il programma "xfontsel". Il programma "xlsfonts" dà una lista di tutti i font disponibili.

Per la GUI Win32 *E244* *E245*

- accetta queste opzioni nel nome del font:
  - hXX - altezza XX (punti, sono accettati i decimali)
  - wXX - larghezza XX (punti, sono accettati i decimali)
  - b - bold (grassetto)
  - i - italic (corsivo)
  - u - underline (sottolineato)
  - s - strikeout (barrato)
  - cXX - tipo-carattere [fontset] XX . Tipi di carattere validi sono: ANSI, ARABIC, BALTIC, CHINESEBIG5, DEFAULT, EASTEUROPE, GB2312, GREEK, HANGEUL, HEBREW, JOHAB, MAC, OEM, RUSSIAN, SHIFTJIS, SYMBOL, THAI, TURKISH, VIETNAMESE ANSI e BALTIC. Normalmente si userebbe "cDEFAULT".
  - qXX - qualità XX. Tipi di qualità validi sono: PROOF, DRAFT, ANTIALIASED, NONANTIALIASED, CLEARTYPE, DEFAULT. Normalmente si userebbe "qDEFAULT". Alcuni valori per la qualità non sono supportati in Sistemi Operativi datati.

  Usate i ':' per separare le opzioni.
- Un '_' si può usare al posto di uno spazio, per non dover usare i backslash per segnalare gli spazi.
- Ad es.: >
```
:set guifont=courier_new:h12:w5:b:cRUSSIAN
:set guifont=Andale_Mono:h7.5:w4.5
```
< Vedere anche |**font-sizes**|.

'guifontset' *'gfs'*
E250 *E252* *E234* *E597* *E598*

'guifontset' 'gfs' stringa (default: "")

globale
{non in Vi}
{disponibile solo se compilato con la GUI
abilitata, e con la funzionalità |**+xfontset**|}
{non disponibile nella GUI GTK+}
Se è impostato, specifica due (o più) font da usare. Il primo per la lingua inglese, e il secondo per la vostra lingua particolare. Vedere |**xfontset**|.
L'impostazione di quest'opzione implica anche che tutti i nomi di font verranno trattati come nomi di un tipo-carattere [fontset]. Anche quelli usato come argomento "font" nel comando |**:highlight**|. I font devono essere in accordo con le impostazioni della lingua locale. Se i font necessari per il set di caratteri della lingua locale non vengono inclusi, impostare 'guifontset' non funzionerà. Nota Differenza fra 'guifont' e 'guifontset': In 'guifont' i nomi separati da virgola sono nomi alternativi, uno solo dei quali verrà usato. In 'guifontset' l'intera stringa è il nome di un tipo-carattere [fontset], virgole incluse. Non si possono specificare dei nomi di tipo-carattere alternativi.
Questo esempio va bene in molti sistemi X11: >
:set guifontset=-*-*-medium-r-normal--16-*-*-*-c-*-*-*
<

'guifontwide' *'gfw'* *E231* *E533* *E534*
'guifontwide' 'gfw' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la GUI abilitata}
Se è impostato, specifica una lista, separata da virgola, di font da usare per caratteri di grandezza doppia. Il primo font che è possibile caricare viene usato.
Nota: La larghezza di questi font deve essere esattamente il doppio di quella specificata in 'guifont' mentre l'altezza deve essere uguale.

Per tutte le versioni GUI tranne GTK+:

'guifontwide' è usato solo quando 'encoding' è impostato a "utf-8" e 'guifontset' è nullo o non valido.
Quando 'guifont' è impostato ed un font valido viene trovato e 'guifontwide' non è impostata, Vim cercherà di trovare un font corrispondente di larghezza doppia, e imposterà 'guifontwide' col valore del font trovato.

Solo per la GUI GTK+: *guifontwide_gtk*

Se impostato e valido, 'guifontwide' è sempre usato per caratteri a larghezza doppia, anche se 'encoding' non è impostato a "utf-8". Vim non cerca di trovare un valore appropriato per 'guifontwide' automaticamente. Se 'guifontwide' non è impostato Pango/Xft sceglierà il font da usare per caratteri non disponibili in 'guifont'. Quindi non è assolutamente necessario impostare 'guifontwide' se non per modificare la scelta che verrebbe fatta da Pango/Xft.

Solo in Windows +multibyte : *guifontwide_win_mbyte*

Se impostato e valido, si usa per IME (Input Method Editor) 'guifontwide' invece che 'guifont'.

'guiheadroom' *'ghr'*
'guiheadroom' 'ghr' numero (default: 50)
globale
{non in Vi} {solo per le GUI GTK e X11}
Il numero di pixel da sottrarre all'altezza dello schermo quando si posiziona la finestra GUI sullo schermo. Da impostare prima della inizializzazione della GUI ad es., nel vostro file |**gvimrc**|. Se a zero, l'intera altezza dello schermo verrà usata per la finestra. Se positivo, il numero specificato di righe di pixel verrà lasciato per le decorazioni della finestra ed altri elementi dello schermo. Impostatelo ad un valore negativo per creare finestre più alte dello schermo.

'guioptions' *'go'*
'guioptions' 'go' stringa (default: "egmrLtT" (MS-Windows, "t" è

rimosso in |**defaults.vim**|),
"aegimrLtT" (GTK, Motif e Athena))
globale
{non in Vi}
{disponibile solo se compilato con la GUI abilitata}
Quest'opzione ha effetto solo nella versione GUI di Vim. È una sequenza di lettere che descrive quali componenti e opzioni della GUI dovrebbero essere usati.
Per evitare problemi con flag che potrebbero essere aggiunti in futuro usate la forma "+=" e "-=" del comando ":set" |**add-option-flags**|.

I caratteri validi sono i seguenti:

'go-!'
'!' I comandi esterni sono eseguiti in una finestra di terminale. Senza questo flag, la GUI di MS-Windows apre una finestra di console per eseguire il comando. La GUI di Unix simula un terminale stupido, per contenere l'output del comando.
La finestra di terminale sarà posizionata a fondo schermo, e crescerà verso l'alto secondo necessità.

guioptions_a *'go-a'*
'a' Autoselect: Se presente, quando il modo VISUAL è iniziato o l'area Visual viene estesa, Vim tenta di diventare il possessore della selezione globale del sistema che gestisce le finestre. Questo vuol dire che il testo evidenziato è disponibile per essere "incollato" in altre applicazioni, come pure nello stesso Vim. Quando il modo Visual termina, magari per una operazione fatta sul testo, o quando una applicazione vuole "incollare" la selezione, il testo è automaticamente messo nel registro di selezione "* .
Quindi il testo selezionato resta disponibile per essere "incollato" in altre applicazioni, anche dopo che il modo VISUAL è terminato.
Se non presente, Vim non diventerà il possessore della selezione globale del sistema di gestione delle finestre, a meno che questo non venga esplicitamente richiesto da un comando di copiatura nel o di cancellatura dal registro "* Lo stesso si applica alla selezione quando si lavora in modo Modeless.

'go-P'
'P' Come autoselect ma utilizzando il registro "+ invece del registro "*.

'go-A'
'A' Autoselect per la selezione modeless. Come 'a', ma si applica solo alla selezione modeless.

'guioptions'	autoselect Visual	autoselect modeless ~
""	-	-
"a"	sì	sì
"A"	-	sì
"aA"	sì	sì

'go-c'
'c' Usare dialoghi di console invece che dialoghi di tipo dinamico (popup) per far effettuare semplici scelte all'utilizzatore.

'go-e'
'e' Aggiungi pagine di linguette quando indicato da 'showtabline'. Si può usare 'guitablabel' per cambiare il testo nelle etichette.
Quando 'e' manca si può usare una riga di demarcazione linguette non-GUI.
Le linguette GUI sono supportate solo su alcuni sistemi, finora GTK, Motif, Mac OS/X e MS-Windows.

'go-f'
'f' Foreground: Non usare fork() per rendere la GUI indipendente dalla shell in cui è stata fatta partire. Da usare per programmi che devono aspettare che la sessione Vim sia finita (ad esempio un programma e-mail). Alternativamente, si può usare "gvim -f" o ":gui -f" per far partire la GUI in foreground. |**gui-fork**|
Nota: Quest'opzione va impostata nel file vimrc. Il fork può già essere stato innescato quando si legge il file |**gvimrc**|.

'go-i'
'i' Usare un'icona Vim. Per il GTK con KDE è usata nell'angolo in alto a sinistra della finestra. È in bianco e nero, se non in GTK, a causa delle limitazioni di X11. Per una icona colorate, vedere |**X11-icon**|.

'go-m'
'm' È presente la barra dei menù.

'go-M'
'M' Il menù di sistema "$VIMRUNTIME/menu.vim" non è utilizzato. Nota: questo flag va aggiunto nel file .vimrc, prima di attivare il riconoscimento della sintassi o del tipo file (quando il file |**gvimrc**| viene eseguito, il menù di sistema è già stato caricato; anche i comandi `:syntax on` e `:filetype on` caricano questo menù).

'go-g'
'g' Elementi di menù grigi: gli elementi di menù che non sono disponibili sono visualizzati in grigio. Se 'g' non è presente, gli elementi di menù inattivi non sono visualizzati per nulla.
Eccezione: la GUI Athena usa menù grigi in ogni caso.

'go-t'
't' Mettere a disposizione elementi di menù "tearoff" [che restano presenti in una parte dello schermo, a prescindere dal menù principale]. Al momento possibile solo per le GUI Win32, GTK+, e Motif 1.2 .

'go-T'
'T' Includere Toolbar [barra degli strumenti].
Al momento possibile solo per le GUI Win32, GTK+, Motif, Photon e Athena.

'go-r'
'r' Barra di scorrimento di destra sempre presente.

'go-R'
'R' Barra di scorrimento di destra presente quando c'è una finestra divisa verticalmente.

'go-l'
'l' Barra di scorrimento di sinistra sempre presente.

'go-L'
'L' Barra di scorrimento di sinistra presente quando c'è una finestra divisa verticalmente.

'go-b'
'b' Barra di scorrimento di fondo (orizzontale) presente. la sua dimensione dipende dalla riga più lunga visualizzabile, o da quella della riga in cui si trova il cursore se è specificata il flag 'h' |**gui-horiz-scroll**|.

'go-h'
'h' Stabilire la dimensione della barra orizzontale in base alla lunghezza della riga in cui si trova il cursore. Questo semplifica i calcoli |**gui-horiz-scroll**|.

E, certo, si possono avere barre di scorrimento sia a sinistra CHE a destra, se veramente si vuole :-). Vedere |**gui-scrollbars**| per ulteriori informazioni.

'go-v'
'v' Usare una disposizione verticale dei pulsanti [di scelta] per i dialoghi. Se non inclusa, una disposizione orizzontale è preferita, ma ne non c'è spazio, ne viene usata una verticale in ogni caso.

'go-p'
'p' Usare "Pointer callback" per la GUI X11. Questo è necessario per alcune gestioni di finestra. Se il cursore non è intermittente o è sfuggente, provate ad aggiungere questa flag. Questo va fatto prima della inizializzazione della GUI. Va impostato nel vostro |**gvimrc**|. Aggiungere il flag o rimuoverlo dopo la partenza della GUI non ha effetto.

'go-F'
'F' Aggiungere un piè di pagina. Solo per Motif. Vedere |**gui-footer**|.

'go-k'
'k' Mantenere la dimensione di finestra della GUI quando si aggiunge/toglie una barra di scorrimento, o una toolbar

[barra degli strumenti], una tabline [barra delle linguette], etc. In questo caso, il comportamento è simile a quello che si ha quando la finestra viene massimizzata e quindi 'lines' e 'columns' sono modificate per adeguarsi alle nuove dimensioni della finestra. Senza il flag 'k' Vim tenta di non variare 'lines' e 'columns' quando si aggiungono o tolgono degli elementi della GUI.

'guipty' *'noguipty'*

'guipty' booleana (default: on)
globale
{non in Vi}
{disponibile solo se compilato con la GUI abilitata}

Solo nella GUI: Se impostato, si tenta di aprire una pseudo-tty per effettuare I/O [input/Output] verso/da comandi shell. Vedere |**gui-pty**|.

'guitablabel' *'gtl'*

'guitablabel' 'gtl' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la GUI abilitata, e con la funzionalità |**+windows**|}

Se diversa dalla stringa nulla, descrive il testo da usare in un'etichetta della riga delle linguette della GUI. Quando è nulla e quando il risultato della valutazione è la stringa nulla, Vim userà un'etichetta di default. Vedere |**setting-guitablabel**| per maggiori informazioni.

Il formato di quest'opzione è simile a quello di 'statusline'. 'guitabtooltip' è usata per dare un suggerimento (tooltip), vedere più sotto.
L'espressione sarà valutata nel |**sandbox**|, se è stata impostata da una modeline; vedere |**sandbox-option**|.

Usata solo quando la riga delle linguette della GUI è visualizzata. 'e' deve essere presente in 'guioptions'. Per la line delle linguette in ambiente non-GUI si usa 'tabline'.

'guitabtooltip' *'gtt'*

'guitabtooltip' 'gtt' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la GUI abilitata, e con la funzionalità |**+windows**|}

Se diversa dalla stringa nulla, descrive il testo da usare come suggerimento in un'etichetta della riga delle linguette della GUI. Quando è nulla Vim userà un suggerimento di default.
Quest'opzione è per il resto come 'guitablabel' più sopra.
Potete includere degli "a capo". Il metodo più semplice è di usare |**:let**|: >

```
:let &guitabtooltip = "riga uno\nriga due"
```

<

'helpfile' *'hf'*

'helpfile' 'hf' stringa (default (MSDOS) "$VIMRUNTIME\doc\help.txt"
(altri) "$VIMRUNTIME/doc/help.txt")
globale
{non in Vi}

Nome del file di aiuto principale. Tutti i file di aiuto disponibili dovrebbero essere messi insieme, in una sola directory. In aggiunta a questo, tutte le directory "doc" contenute in 'runtimepath' saranno usate.
Le variabili d'ambiente vengono valutate |**:set_env**|. Ad esempio: "$VIMRUNTIME/doc/help.txt". Se $VIMRUNTIME non è impostato, si tenta anche di utilizzare $VIM.
Vedere anche |**$VIMRUNTIME**| e |**option-backslash**| per inserire spazi e backslash.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'helpheight' *'hh'*

'helpheight' 'hh' numero (default: 20)

globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+windows**|}
Altezza iniziale minima della finestra di aiuto quando è aperta con il comando ":help". L'altezza iniziale della finestra di aiuto è di metà della finestra corrente, o (quando l'opzione 'ea' è a on) della stessa dimensione delle altre finestre. Quando l'altezza delle finestra è minore di 'helpheight', l'altezza stessa viene portata a 'helpheight'. Impostare a zero per disabilitare l'opzione.

'helplang' *'hlg'*
'helplang' 'hlg' stringa (default: lingua per i messaggi o stringa nulla)
globale
{disponibile solo se compilato con la funzionalità
|**+multi_lang**|}
{non in Vi}
Lista di lingue, separate da virgola. Vim utilizzerà la prima lingua per la quale il file di aiuto desiderato è disponibile. L'aiuto in inglese sarà sempre utilizzato se non è possibile fare diversamente. Si può aggiungere "en" per scegliere l'inglese al posto di un altra lingua, ma in questo modo verranno trovate solo le tag che esistono esclusivamente per quella lingua, e non nei testi di aiuto in inglese.
Esempio: >
:set helplang=de,it
< In questo modo verranno cercate le pagine di aiuto in tedesco, poi quelle in italiano, e da ultimo quelle in inglese.
Quando si usa |**CTRL-]**| e ":help!" in un file di aiuto che non sia in inglese, Vim tenta di trovare il tag nella lingua corrente, prima di usare quest'opzione.
Vedere |**help-translated**|.

'hidden' *'hid'* *'nohidden'* *'nohid'*
'hidden' 'hid' booleana (default: off)
globale
{non in Vi}
Se inattiva un buffer viene scaricato quando è lasciato |**abandon**|.
Quando è attiva un buffer diviene nascosto [hidden] quando è lasciato |**abandon**|. Se il buffer è ancora visualizzato in un'altra finestra, non diventa nascosto, naturalmente.
I comandi che permettono di muoversi in una buffer-list talora rendono nascosto un buffer, anche se l'opzione 'hidden' è a off: quando il buffer è modificato, 'autowrite' è a off o la scrittura non è possibile, ed è stata usata il flag '!'. Vedere anche |**windows.txt**|.
Per nascondere solo un buffer, usare l'opzione 'bufhidden'.
Quest'opzione è impostata per un comando on ":hide {command}" |**:hide**|.
ATTENZIONE: È facile scordarsi che ci sono modifiche in buffer nascosti. Pensateci due volte, quando usate ":q!" o ":qa!".

'highlight' *'hl'*
'highlight' 'hl' stringa (default (un'unica stringa):
"8:SpecialKey,~:EndOfBuffer,@:NonText,
d:Directory,e:ErrorMsg,i:IncSearch,
l:Search,m:MoreMsg,M:ModeMsg,n:LineNr,
N:CursorLineNr,r:Question,s:StatusLine,
S:StatusLineNC,c:VertSplit,t:Title,
v:Visual,w:WarningMsg,W:WildMenu,f:Folded,
F:FoldColumn,A:DiffAdd,C:DiffChange,
D:DiffDelete,T:DiffText,>:SignColumn,
B:SpellBad,P:SpellCap,R:SpellRare,
L:SpellLocal,-:Conceal,+:Pmenu,=:PmenuSel,
x:PmenuSbar,X:PmenuThumb,*:TabLine,
#:TabLineSel,_:TabLineFill,!:CursorColumn,
.:CursorLine,o:ColorColumn,q:QuickFixLine,
z:StatusLineTerm,Z:StatusLineTermNC")
globale
{non in Vi}
Quest'opzione si può usare per impostare il modo di evidenziazione per varie situazioni. È un elenco separato da virgola di coppie di

caratteri. Il primo carattere della coppia descrive la situazione, il secondo la modalità da usare in quella situazione. Le situazioni sono:

	hl-SpecialKey	8	tasti Meta e Speciali listati con ":map"
	hl-EndOfBuffer	~	righe dopo l'ultima riga nel buffer
	hl-NonText	@	'@' a fine finestra, e caratteri da 'showbreak'
	hl-Directory	d	directory in liste CTRL-D e altre cose speciali nelle liste
	hl-ErrorMsg	e	Messaggi di errore
		h	(obsoleto, ignorare)
	hl-IncSearch	i	evidenziazione per 'incsearch'
	hl-Search	l	evidenziazione espressione di ricerca usata per ultima (vedere 'hlsearch')
|**hl-MoreMsg**| m	richiesta di continuare una visualizzazione. Vedere |**more-prompt**|
	hl-ModeMsg	M	modo (ad es., "-- INSERT --")
	hl-LineNr	n	numero di riga per comandi ":number" e ":#", e per quando sia impostata l'opzione 'number' o 'relativenumber'.
	hl-CursorLineNr	N	come n per quando 'cursorline' o 'relativenumber' è impostata.
|**hl-Question**| r	richiesta di dare <INVIO> |**hit-enter**| e domande di conferma (yes/no)
	hl-StatusLine	s	riga di status della finestra corrente
	hl-StatusLineNC		S	righe di stato per le finestre non correnti
	hl-Title	t	titoli per output di ":set all", ":autocmd" etc.
	hl-VertSplit	c	colonna usata per separare finestre separate verticalmente
	hl-Visual	v	modo Visual
|**hl-VisualNOS**| V	modo Visual quando Vim non controlla la selezione ["Not Owning the Selection"] Solo per clipboard GUI X11 |**gui-x11**| e |**xterm-clipboard**|.
	hl-WarningMsg	w	messaggi di avviso
	hl-WildMenu	W	corrispondenze trovate, visualizzate per 'wildmenu'
	hl-Folded	f	riga usata per indicare piegatura chiusa
	hl-FoldColumn	F	'foldcolumn'
	hl-DiffAdd	A	riga aggiunta in modo Diff
	hl-DiffChange	C	riga modificata in modo Diff
	hl-DiffDelete	D	riga cancellata in modo Diff
	hl-DiffText	T	testo inserito in modo Diff
|**hl-SignColumn**| >	colonna usata per indicazioni |**signs**|
|**hl-SpellBad**| B	parola scritta male |**spell**|
|**hl-SpellCap**| P	parola che dovrebbe iniziare con una lettera maiuscola |**spell**|
|**hl-SpellRare**| R	parola inconsueta |**spell**|
|**hl-SpellLocal**| L	parola non usata localmente |**spell**|
	hl-Conceal	-	il segnaposto usato per caratteri nascosti (vedere 'conceallevel')
	hl-Pmenu	+	menù dinamico (popup), riga normale
	hl-PmenuSel	=	menù dinamico, riga normale
	hl-PmenuSbar	x	menù dinamico, barra scorrimento
	hl-PmenuThumb	X	menù dinamico, guida per barra scorrimento

I modi di visualizzazione sono:

r	invertito (reverse)	(elementi termcap "mr" e "me")
i	corsivo (italic)	(elementi termcap "ZH" e "ZR")
b	grassetto (bold)	(elementi termcap "md" e "me")
s	rilievo (standout)	(elementi termcap "so" e "se")
u	sottolineato (underline)	(elementi termcap "us" e "ue")
c	sottolineato con tilde, errore (undercurl)	(elementi termcap "Cs" e "Ce")
t	barrato (elemento termcap "Ts" e "Te")	
n	nessuna evidenziazione	
-	nessuna evidenziazione	
:	usa un gruppo di evidenziazione	

Il default è usato per le situazioni non presenti nella lista.
Per cambiare quello che fanno i modi di visualizzazione, vedere |**dos-colors**| per avere un esempio.
Quando si usa id modo di visualizzazione ':', i ":" devono essere seguiti dal nome di un gruppo di evidenziazione. Un gruppo di evidenziazione si può usare per definire qualsiasi tipo di

evidenziazione, compresa la colorazione. Vedere |**:highlight**| per come definirne uno. Il default usa un gruppo differente per ogni situazione. Vedere |**highlight-default**| per i gruppi di visualizzazione di default.

'history' *'hi'*
'history' 'hi' numero (default Vim: 50, default Vi: 0, impostato a 200 in |**defaults.vim**|)
globale
{non in Vi}

La storia dei comandi ":", e la storia delle precedenti espressioni di ricerca viene memorizzata. Quest'opzione specifica quante righe possono essere immagazzinate in ognuna di queste storie (vedere |**cmdline-editing**|).
Il valore massimo è 10000.
NOTA: Quest'opzione è impostata al valore di default di Vi impostando 'compatible' e al valore di default di Vim quando 'compatible' viene messa a off.

'hkmap' *'hk'* *'nohkmap'* *'nohk'*
'hkmap' 'hk' booleana (default: off)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+rightleft**|}

Se impostata, la tastiera è mappata per scrivere in Ebraico.
Normalmente si imposta anche 'allowrevins' e si usa CTRL-_ in modo Insert per abilitare/disabilitare quest'opzione. Vedere |**rileft.txt**|.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'hkmapp' *'hkp'* *'nohkmapp'* *'nohkp'*
'hkmapp' 'hkp' booleana (default: off)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+rightleft**|}

Se impostata, la mappatura fonetica della tastiera è usata. 'hkmap' deve anche essere impostato a on.
Questo è utile se sia ha una tastiera non Ebraica.
Vedere |**rileft.txt**|.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'hlsearch' *'hls'* *'nohlsearch'* *'nohls'*
'hlsearch' 'hls' booleana (default: off)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+extra_search**|}

Quando già esiste una espressione di ricerca, evidenzia tutte le sue corrispondenze. Il tipo di evidenziazione può essere impostato con la situazione 'l' dell'opzione 'highlight'. Questa usa per default il gruppo di evidenziazione "Search". Nota Solo il testo corrispondente è evidenziato, alcune corrispondenze non sono applicate [se ci dovessero essere più corrispondenze, la seconda che inizia prima che finisca l'altra].
Vedere anche: 'incsearch' e |**:match**|.
Se vi annoia vedere le corrispondenze evidenziate, potete evitarlo dando |**:nohlsearch**|. Questo non cambia il valore dell'opzione: non appena si usa nuovo un comando di ricerca, l'evidenziazione ritornerà attiva.
'redrawtime' specifica il tempo massimo impiegato per trovare corrispondenze.
Quando l'espressione di ricerca comprende una fine-riga, Vim cerca di evidenziare tutto il testo corrispondente. Questo, comunque, dipende da dove la ricerca ha inizio. Questa sarà la prima riga visibile nella finestra o la prima riga sotto una piegatura chiusa. Una corrispondenza in una riga precedente e che non è visibile sullo schermo può non continuare in una riga di quelle presenti sullo schermo [ossia, la fine di una corrispondenza "precedente" può non essere visualizzata, prima della corrispondenza corrente].
È possibile specificare che la scelta di evidenziazione sia ripristinata in una nuova sessione Vim mediante la flag 'h' in 'viminfo' |**viminfo-h**|.

NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'icon' *'noicon'*
'icon' booleana (inattiva per default, attiva se il titolo può essere controllato)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |+title|}

Se impostata, il testo della finestra ridotta a icona sarà impostato al valore di 'iconstring' (se non è nullo) o al nome del file in modifica in quel momento. Solo l'ultima parte del nome viene usata. Può essere ulteriormente modificato dall'opzione 'iconstring'. Funziona solo se il terminale supporta l'impostazione di icone di finestra (per ora disponibili solo per la GUI X11 e per terminali con un'opzione 't_IS' non nulla - ossia Xterm Unix e iris-ansi per default - per i quali 't_IS' è presa dalle termcap [descrizioni delle proprietà del terminale] native).
Se Vim è stato compilato con il flag HAVE_X11, l'icona originale verrà ripristinata se possibile, |**X11**|. Vedere |**X11-icon**| per cambiare icona in X11.
Per MS-Windows è possibile cambiare icona, vedere |**windows-icon**|.

'iconstring'
'iconstring' stringa (default: "")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |+title|}

Quando quest'opzione non è nulla, sarà usata per il testo dell'icona della finestra. Questo avviene solo se l'opzione 'icon' è impostata. Funziona solo se il terminale supporta l'impostazione di un testo per l'icona della finestra (per ora solo la GUI X11 e terminali con un'opzione 't_IS' non nulla).
Non funziona per MS Windows.
Se Vim è stato compilato con il flag HAVE_X11, l'icona originale verrà ripristinata se possibile, |**X11**|.
Quando quest'opzione contiene elementi '%' sul tipo di quelli di "printf", saranno valorizzati con le stesse regole usate per 'statusline'. Vedere 'titlestring' per esempi di impostazione.
{non disponibile se compilato senza la funzionalità |**+statuslines**|}

'ignorecase' *'ic'* *'noignorecase'* *'noic'*
'ignorecase' 'ic' booleana (default: off)
globale

Ignora maiuscolo/minuscolo nelle espressioni regolari di ricerca. Usato anche se si fanno ricerche nel file tags.
Vedere anche 'smartcase' e 'tagcase'.
Si può inattivare temporaneamente usando "\c" o "\C" nell'espressione, vedere|**/ignorecase**|.

'imactivatefunc' *'imaf'*
'imactivatefunc' 'imaf' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+mbyte**|}

Quest'opzione specifica una funzione da chiamare per attivare/ disattivare per attivare o disattivare l'Input Method [Metodo di Input].
Non è usata in ambiente GUI.

Esempio: >

```
function ImActivateFunc(active)
  if a:active
    ... fai qualcosa
  else
    ... fai qualcosa
  endif
  " il valore ritornato dalla funzione non è usato
endfunction
set imactivatefunc=ImActivateFunc
```

<

'imactivatekey' *'imak'*
'imactivatekey' 'imak' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con |**+xim**| e
|**+GUI_GTK**|} *E599*
Specifica il tasto che il vostro Metodo di Input in X-Windows usa per l'attivazione. Quando si specifica questo correttamente, vim può controllare completamente l'Input Method [Metodo di Input] con le opzioni 'imcmdline', 'iminsert' e 'imsearch'.
Quest'opzione NON può essere usata per cambiare il tasto di attivazione, l'opzione si limita a dire a vim di che tasto si tratta.
Formato:
[MODIFIER_FLAG-]KEY_STRING

I seguenti caratteri si possono usare per MODIFIER_FLAG (maiuscolo/minuscolo è ignorato):

S	tasto Maiuscolo [Shift]
L	tasto Blocca Maiuscolo [Lock]
C	tasto Control [Ctrl]
1	tasto Mod1
2	tasto Mod2
3	tasto Mod3
4	tasto Mod4
5	tasto Mod5

Combinazioni sono permesse, ad es. "S-C-space" o "SC-space" stanno a indicare entrambi maiuscolo+control+spazio.
Vedere <X11/keysymdef.h> e XStringToKeysym per KEY_STRING.

Esempio: >
:set imactivatekey=S-space
< "S-space" sta per maiuscolo+spazio. Questa è la chiave di attivazione per kinput2 + canna (Giapponese), e ami (Coreano).

'imcmdline' *'imc'* *'noimcmdline'* *'noimc'*
'imcmdline' 'imc' booleana (default: off)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|**+mbyte**|}
Se impostata, il Metodo di Input è sempre attivo quando si inizia a modificare una riga di comando, a meno che non si introduca un'espressione regolare da ricercare (vedere 'imsearch' al riguardo).
Impostare quest'opzione è utile quando il vostro metodo di Input consente di inserire caratteri inglesi direttamente, ad es., quando lo si usi per inserire caratteri accentati usando dei tasti "morti" (ossia che restano sul carattere che si vuole modificare).

'imdisable' *'imd'* *'noimdisable'* *'noimd'*
'imdisable' 'imd' booleana (inattiva per default, attiva per alcuni sistemi (SGI))
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|**+mbyte**|}
Se impostata, il Metodo di Input non è mai usato. Ciò può tornare utile per disabilitare l'IM [Metodo di Input] quando non funziona come dovrebbe.
Attualmente quest'opzione è attivata per default su macchine SGI/IRIX. Questo potrà cambiare in future versioni.

'iminsert' *'imi'*
'iminsert' 'imi' numero (default: 0)
locale al buffer
{non in Vi}
Specifica se :lmap o un IM [Metodo di Input] va usato in modo Insert.
Valori ammessi:

0	:lmap inattiva e IM inattivo
1	:lmap ATTIVA e IM inattivo
2	:lmap inattiva e IM ATTIVO

Per reimpostare sempre a zero l'opzione quando si esce dal modo Insert con <Esc> si può usare: >
:inoremap <ESC> <ESC>:set iminsert=0<CR>

```
<       Questo inattiva :lmap e IM automaticamente all'uscita dal modo
        Insert.
        Nota Quest'opzione viene modificata se si usa CTRL-^ in modo Insert
        |i_CTRL-^|.
        Il valore è impostato a 1 quando si imposta 'keymap' a un nome valido
        di "keymap" [mappatura tastiera].
        Usato anche come argomento di comandi come "r" e "f".
        Il valore 0 può non funzionare correttamente con le GUI Athena e Motif
        e con alcuni XIM [Metodi di Input X].  Usare 'imdisable' per
        disabilitare l'XIM in questo caso.

        Si possono impostare 'imactivatefunc' e 'imstatusfunc' per gestire
        IME/XIM attraverso un comando esterno, nel caso in cui vim non sia
        stato compilato con le opzioni |+xim|, |+multi_byte_ime| o |global-ime|.

                                                *'imsearch'* *'ims'*
'imsearch' 'ims'        numero  (default: -1)
                        locale al buffer
                        {non in Vi}
        Specifica se :lmap o un IM [Metodo di Input] va usato quando si
        immette un'espressione di ricerca.  Valori ammessi:
                -1      si usa il valore di 'iminsert', ossia equivale a usare
                        'iminsert' anche quando si immette un'espressione di
                        ricerca
                0       :lmap inattiva e IM inattivo
                1       :lmap ATTIVA e IM inattivo
                2       :lmap inattiva e IM ATTIVO
        Nota Quest'opzione viene modificata se si usa CTRL-^ in modo
        Command-line |c_CTRL-^|.
        Il valore è impostato a 1 se non è -1 e se si imposta l'opzione
        'keymap' a un nome valido di keymap [mappatura tastiera].
        Il valore 0 può non funzionare correttamente con le GUI Athena e Motif
        e con alcuni XIM [Metodi di Input X].  Usare 'imdisable' per
        disabilitare l'XIM in questo caso.

                                                *'imstatusfunc'* *'imsf'*
'imstatusfunc' 'imsf'   stringa (default: "")
                        globale
                        {non in Vi}
                        {disponibile solo se compilato con la funzionalità
                        |+mbyte|}
        Quest'opzione specifica una funzione da chiamare per ottenere lo
        status dell'Input Method [Metodo di Input].  Deve ritrornare un
        numero positivo quando l'IME [Input Method Editor] è attivo.
        Non è usata in ambiente GUI.

        Esempio: >
                function ImStatusFunc()
                  let is_active = ...fai qualcosa
                  return is_active ? 1 : 0
                endfunction
                set imstatusfunc=ImStatusFunc
<
        NOTA: Questa funzione è chiamata molto spesso.  Deve essere veloce.

                                                *'imstyle'* *'imst'*
'imstyle' 'imst'        numero (default: 1)
                        globale
                        {non in Vi}
                        {disponibile solo se compilato con le funzionalità
                        |+xim| e |+GUI_GTK|}
        Quest'opzione specifica lo stile di input del Metodo di Input
        [Input Method].
        0  usare lo stile 'on-the-spot' [in-questo-punto].
        1  usare lo stile 'over-the-spot' [sopra-questo-punto].
        Vedere: |xim-input-style|

        Per molto tempo è stato in uso, nella versione GTK di Vim, lo stile
        'on-the-spot'; è però accertato che ciò può causare problemi quando
        si usino mappature, |single-repeat|, etc.  Per questo motivo lo stile
        'over-the-spot' è il nuovo default.  Questo valore di default
        dovrebbe andar bene per la maggioranza degli utenti.  Se si
        incontrano problemi usandolo, si provi a usare lo stile
```

'on-the-spot'.

'include' *'inc'*
'include' 'inc' string (default: "^\s*#\s*include")
globale o locale al buffer |**global-local**|
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+find_in_path**|}
Espressione di ricerca da usare per trova un comando di "include". E' un'espressione di ricerca, proprio come per il comando "/" (Vedere |**pattern**|). Il default è quello adatto ai programmi C. Questa opzione è usata per i comandi "[i", "]I", "[d", etc.
Normalmente l'opzione 'isfname' è usata per riconoscere il nome di file che segue l'espressione che è stata individuata. Ma se "\zs" è incluso nell'espressione, allora il testo a partire da "\zs" fino alla fine, o finché non si arriva a "\ze", se presente, è usato come nome del file. Da usare per includere caratteri che non sono parte di 'isfname', tipo lo spazio. Si può poi usare 'includeexpr' per gestire il testo così trovato.
Vedere |**option-backslash**| per inserire spazi e backslash.

'includeexpr' *'inex'*
'includeexpr' 'inex' stringa (default: "")
locale al buffer
{non in Vi}
{non disponibile se compilato senza le funzionalità
|**+find_in_path**| o |**+eval**|}
Espressione da usare per ricavare dalla stringa trovata con l'opzione 'include' un nome di file. Usato soprattutto per cambiare "." in "/" in Java: >
:set includeexpr=substitute(v:fname,'\\.','/','g')
< La variabile "v:fname" sarà impostata al nome file così ricavato.

Usata anche per il comando |**gf**| se non si riesce a trovare un nome file non modificato. Questo consente di eseguire "gf" sul nome che segue un'istruzione 'include'.
Usato anche per |**<cfile>**|.

L'espressione sarà valutata nel |**sandbox**|, se è stata impostata da una modeline; vedere |**sandbox-option**|.

Non è consentito cambiare testo o passare a un'altra finestra mentre è in corso la valutazione di 'includeexpr' |**textlock**|.

'incsearch' *'is'* *'noincsearch'* *'nois'*
'incsearch' 'is' booleana (default: off)
'incsearch' 'is' booleana (default: off, impostata a on in
|**defaults.vim**| se la
funzionalità +reltime è
disponibile)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+extra_search**|}
Mentre si sta immettendo un comando di ricerca, mostra immediatamente dove si incontra l'espressione che si è immessa fino a quel momento. La stringa così individuata è evidenziata. Se l'espressione non è valida o non è trovata, non viene mostrato nulla. Lo schermo viene aggiornato spesso, quindi quest'opzione è utilizzabile solo su terminali veloci.
Si noti che la stringa individuata viene evidenziata, ma che il cursore ritornerà nella posizione originale se non viene trovata alcuna corrispondenza, e quando si immette <Esc>. Occorre concludere il comando di ricerca, dando <Invio>, per far spostare il cursore sulla corrispondenza.
Si può usare la combinazione di tasti CTRL-G e CTRL-T per spostarsi sulla corrispondenza successiva e su quella precedente.
Si può usare la combinazione di tasti CTRL-N e CTRL-P per spostarsi sulla corrispondenza successiva e su quella precedente.
|**c_CTRL-G**| |**c_CTRL-T**|
Quando compilato con la funzionalità |**+reltime**|, Vim ricerca solo per circa mezzo secondo. Con un'espressione di ricerca complicata e se il testo in modifica è molto grande, la corrispondenza può non

essere trovata (in tempo utile). Ciò per evitare che Vim si blocchi mentre voi state ancora digitando l'espressione di ricerca.
L'evidenziazione può essere impostata con il flag 'i' dell'opzione 'highlight'.
Quando 'hlsearch' è attivo, tutte le stringhe corrispondenti sono evidenziate anche mentre si sta digitando un comando di ricerca.
Vedere anche: 'hlsearch'.
Se non si vuol attivare 'hlsearch', ma si vogliono evidenziare tutte le corrispondenze mentre si effettua una ricerca, si può disattivare e attivare 'hlsearch' con un autocomando.
Esempio: >

```
augroup vimrc-incsearch-highlight
  autocmd!
  autocmd CmdlineEnter /\? :set hlsearch
  autocmd CmdlineLeave /\? :set nohlsearch
augroup END
```

<

CTRL-L può essere utilizzato per aggiungere alla riga di comando il carattere che segue immediatamente la corrispondenza corrente. Se 'ignorecase' e 'smartcase' sono impostati e la linea dei comandi non contiene caratteri maiuscoli, il carattere aggiunto viene convertito a lettera minuscola.
CTRL-R CTRL-W possono essere utilizzati per aggiungere alla riga di comando la parola alla fine della corrispondenza corrente, dopo i caratteri che sono già stati immessi.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'indentexpr' *'inde'*

'indentexpr' 'inde' stringa (default: "")
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+cindent**| o |**+eval**|}

Espressione da valutare per ottenere l'indentatura per una riga.
Usata quando si crea una nuova riga, con l'operatore |**=**| e in modo Insert come specificato nell'opzione 'indentkeys'.
Quando quest'opzione non è nulla, prevale su quelle specificate per l'indentatura con 'cindent' e 'smartindent'. Quando 'lisp' è impostato, quest'opzione è ignorata, e si usa invece l'algoritmo di indentatura proprio del Lisp.
Quando l'opzione 'paste' è impostata, quest'opzione non è utilizzata per l'indentatura.
L'espressione è valutata con la variabile |**v:lnum**| impostata al numero della riga per la quale l'espressione si deve calcolare. Il cursore è anche su questa riga mentre l'espressione viene calcolata (ma può essere spostato a piacere).
L'espressione deve fornire il numero di spazi di indentatura. Può valere "-1" per indicare di mantenere l'indentatura corrente (ciò significa che 'autoindent' viene usato per l'indentatura).
Funzioni utili per calcolare l'indentatura sono |**indent()**|, |**cindent()**| e |**lispindent()**|.
La valutazione dell'espressione non deve avere effetti secondari! Non deve modificare il testo, saltare a un'altra finestra, etc. Dopo la valutazione, la posizione del cursore è sempre rimessa come era prima, quindi si può muovere il cursore [durante la valutazione].
Normalmente quest'opzione potrà essere impostata per chiamare una funzione: >

```
:set indentexpr=GetMyIndent()
```

< I messaggi di errore non verranno visualizzati, a meno che l'opzione 'debug' contenga "msg".
Vedere |**indent-expression**|.
NOTA: Quest'opzione è impostata a "" se si attiva 'compatible'.

L'espressione sarà valutata nel |**sandbox**|, se è stata impostata da una modeline; vedere |**sandbox-option**|.

Non è consentito cambiare testo o passare a un'altra finestra mentre è in corso la valutazione di 'includeexpr' |**textlock**|.

'indentkeys' *'indk'*

'indentkeys' 'indk' stringa (default: "0{,0},:,0#,!^F,o,O,e")
locale al buffer
{non in Vi}

{non disponibile se compilato senza la funzionalità
|+cindent|}
Una lista di tasti che, quando battuti, provocano una re-indentatura della riga corrente. Funziona solo se 'indentexpr' non è nullo.
Il formato è lo stesso di 'cinkeys', vedere |**indentkeys-format**|.
Vedere |**C-indenting**| e |**indent-expression**|.

'infercase' *'inf'* *'noinfercase'* *'noinf'*
'infercase' 'inf' booleana (default: off)
locale al buffer
{non in Vi}
Mentre si effettua il completamento di parola in modo Insert |**ins-completion**|, e se 'ignorecase' è attivo, il modo di individuare la corrispondenza è adattato, a seconda del testo immesso. Se il testo immesso contiene una lettera minuscola, e la corrispondenza trovata ha una lettera maiuscola, la parte completata è resa minuscola. Se il testo immesso non contiene lettere minuscole e la corrispondenza ha una lettera minuscola dove il testo immesso ha una lettera maiuscola, e c'è una lettera prima di questa, la parte completata viene scritta tutta in maiuscolo.
Con 'noinfercase' la corrispondenza è usata come viene trovata.

'insertmode' *'im'* *'noinsertmode'* *'noim'*
'insertmode' 'im' booleana (default: off)
globale
{non in Vi}
Fa lavorare Vim in modo che il modo Insert sia il modo di lavorare di default. Utile se si vuole usare Vim come un editor senza-modi [tipo Notepad]. Usato per |**evim**|.
Questi comandi saranno utili in modo Insert:
- Usare i tasti che muovono il cursore per spostarsi nel testo.
- Usare CTRL-O per eseguire un comando in modo Normal |**i_CTRL-O**|. Quando questo è mappato, viene eseguito come se 'insertmode' fosse non impostato.
Il modo Normal rimane attivo fino alla fine della mappatura.
- Usare CTRL-L per eseguire dei comandi in modo Normal, e alla fine usare <Esc> per ritornare in modo Insert. Notare che CTRL-L sposta il cursore a sinistra, come fa <Esc> quando 'insertmode' non è impostato- |**i_CTRL-L**|

Questi elementi cambiano quando 'insertmode' è impostato:
- quando si inizia a modificare un file, Vim passa in modo Insert.
- <Esc> in modo Insert non fa nulla tranne un suono di avviso.
- <Esc> in modo Normal fa andare Vim in modo Insert.
- CTRL-L in modo Insert è un comando, non viene inserito nel testo.
- CTRL-Z in modo Insert sospende Vim, vedere |**CTRL-Z**|. *i_CTRL-Z*
Comunque, quando si usa <Esc> all'interno di una mappatura, si comporta come se 'insertmode' non fosse impostato. Il motivo per cui questo accade, è per essere in grado di usare le stesse mappature, tanto con 'insertmode' attivo che inattivo.
Quando si eseguono comandi con |**:normal**| 'insertmode' non è utilizzato.

NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'isfname' *'isf'*
'isfname' 'isf' stringa (default per MS-DOS, Win32 e OS/2:
"@,48-57,/,\,.,-,_,+,,,#,$,%,{,},[,],:,@-@,!,~,="
per AMIGA: "@,48-57,/,.,-,_,+,,,$,:"
per VMS: "@,48-57,/,.,-,_,+,,,#,$,%,<,>,[,],:,;,~"
per OS/390: "@,240-249,/,.,-,_,+,,,#,$,%,~,="
se no: "@,48-57,/,.,-,_,+,,,#,$,%,~,=")
globale
{non in Vi}
I caratteri specificati in quest'opzione sono considerati parte di un nome_file e di un nome di directory. I nomi_file sono usati per comandi come "gf", "[i" e nei file di tag. Si usa anche per "\f" in un'espressione regolare. Vedere |**pattern**|.
Caratteri multi-byte da 256 in su sono sempre inclusi, solo i caratteri fino a 255 sono specificati in quest'opzione.
Per UTF-8 i caratteri 0xa0 fino a 0xff sono pure inclusi.
Pensateci bene prima di aggiungere lo spazio a quest'opzione. Sebbene uno spazio possa apparire all'interno di un nome di file, l'effetto sarebbe che Vim non è in grado di stabilire dove inizia o finisce un

nome di file, mentre effettua il completamento.
È molto probabile che il risultato sia migliore senza inserire uno spazio in 'isfname'.

Nota Nei sistemi che usano il backslash [\] come separatore nel nome directory, Vim tenta di fare del suo meglio per farlo funzionare come ci si aspetta. Questo è abbastanza complicato, perché Vi nella versione originale usava il backslash per proteggere dei caratteri speciali. Vim non rimuove un backslash davanti a un carattere normale che appaia in un nome_file, ma lo fa in Unix e simili. I caratteri '&' e '^' non sono inclusi per default, perché hanno un significato speciale per il programma cmd.exe.

Il formato di quest'opzione è una lista di parti, separate da virgole. Ogni parte può essere un singolo numero di carattere, o una gamma. Una gamma è costituita da due numeri di carattere, separati da '-'. Un numero di carattere può essere un numero fra 0 e 255 o il carattere ASCII stesso (questo non vale per le cifre). Ad es.:

```
"_,-,128-140,#-43"      (include '_' e '-' e le gamme da
                        128 a 140 e da '#' a 43)
```

Se una parte inizia con '^', la gamma o il carattere che seguono saranno esclusi dall'opzione. L'opzione è interpretata da sinistra a destra. Mettete i caratteri da escludere dopo la gamma in cui sono inclusi. Per includere '^' inteso come carattere, lo si usi come ultimo carattere o come fine di una gamma. Ad es.:

```
"^a-z,#,^"      (esclude da 'a' a 'z', include '#' e '^')
```

Se il carattere è '@', tutti i caratteri per i quali la funzione isalpha() restituisce il valore TRUE [vero] sono inclusi. Di solito questo sono i caratteri da a a z e da A a Z, oltre ai caratteri accentati. Per includere '@' inteso come carattere, usate la notazione "@-@". Ad es.:

```
"@,^a-z"        Tutti i caratteri alfabetici, tranne
                le lettere minuscole ASCII.
"a-z,A-Z,@-@"   Tutte le lettere più il carattere '@'.
```

La virgola può essere inclusa usandola al posto di un numero di carattere. Ad es.:

```
"48-57,,,_"     Le cifre, la virgola e il sottolineato.
```

La virgola si può escludere mettendoci prima '^'. Ad es.:

```
" -~,^,,9"      Tutti i caratteri dallo spazio alla tilde,
                tranne la virgola, e con in più il <Tab>.
```

Vedere |**option-backslash**| per inserire spazi e backslash.

'isident' *'isi'*

'isident' 'isi' stringa (default per MS-DOS, Win32 e OS/2:
"@,48-57,_,128-167,224-235"
se no: "@,48-57,_,192-255")
globale
{non in Vi}

I caratteri dati in quest'opzione sono inclusi negli identificatori. Gli identificatori sono usati per riconoscere variabili d'ambiente, e dopo aver trovato una ricorrenza dell'opzione 'define'. È anche usato per "\i" in un'espressione di ricerca |**pattern**|. Vedere 'isfname' per una descrizione del formato di quest'opzione.
Attenzione: Cambiare quest'opzione può impedire una valutazione corretta delle variabili d'ambiente. Ad es., quando si include '/' e vim prova a espandere "$HOME/.viminfo". Forse dovreste cambiare invece 'iskeyword'.

'iskeyword' *'isk'*

'iskeyword' 'isk' stringa (default Vim per MS-DOS e Win32:
"@,48-57,_,128-167,224-235"
se no: "@,48-57,_,192-255"
Default Vi: "@,48-57,_")
locale al buffer
{non in Vi}

Le parole (keywords) sono usate per ricerca e riconoscimento in molti comandi: "w", "*", "[i", etc. L'opzione è anche usata per "\k" in un'espressione regolare |**pattern**|. Vedere 'isfname' per una descrizione del formato di quest'opzione. Per programmi C si potrebbe usare: "a-z,A-Z,48-57,_,.,-,>".
Per un file di aiuto è impostata a tutti i caratteri stampabili, tranne '*', '"' e '|' (così che CTRL-] battuto sopra un comando trovi l'help per quel comando).

Quando l'opzione 'lisp' è attiva, il carattere '-' è sempre incluso.
Quest'opzione ha effetti anche sull'evidenziazione sintattica, a meno
che la sintassi faccia uso di |**:syn-iskeyword**|.
NOTA: Quest'opzione è impostata al valore di default di Vi
impostando 'compatible' e al valore di default di Vim quando
'compatible' viene messa a off.

'isprint' *'isp'*
'isprint' 'isp' stringa (default per MS-DOS, Win32, OS/2 e Macintosh:
"@,~-255"; se no: "@,161-255")
globale
{non in Vi}
I caratteri specificati con quest'opzione sono visualizzati
direttamente sullo schermo. Serve anche quando si specifica "\p"
in un'espressione regolare |**pattern**|. I caratteri dallo spazio
(ASCII 32) fino alla tilde '~' (ASCII 126) sono sempre visualizzati
direttamente, anche se non sono inclusi in 'isprint' o se sono
esclusi. Vedere 'isfname' per una descrizione del formato di questa
opzione.

I caratteri non stampabili sono visualizzati usando due caratteri:

0 - 31	"^@" - "^_"
32 - 126	sempre caratteri singoli
127	"^?"
128 - 159	"~@" - "~_"
160 - 254	"| " - "|~"
255	"~?"

Se 'encoding' è di tipo Unicode, i byte non validi fra 128 e 255 sono
visualizzati come <xx>, con il valore esadecimale del byte.
Quando l'opzione 'display' contiene "uhex" tutti i caratteri non
stampabili sono visualizzati come <xx>.
L'evidenziazione di tipo SpecialKey (caratteri speciali) sarà usata
per i caratteri non stampabili.
|**hl-SpecialKey**|

I caratteri rappresentati con più di un byte, di valore 256 e oltre
sono sempre inclusi, solo i caratteri dal valore fino a 255 sono
specificabili con quest'opzione. Quando un carattere è stampabile
ma non è disponibile nel font che si sta utilizzando, un carattere di
rimpiazzo verrà visualizzato.
Caratteri Unicode non stampabili e di larghezza zero sono visualizzati
come <xxxx>.
Non esiste un'opzione per specificare questi caratteri.

'joinspaces' *'js'* *'nojoinspaces'* *'nojs'*
'joinspaces' 'js' booleana (default: on)
globale
{non in Vi}
Inserire due spazi dopo un '.', '?' e '!' quando si esegue un comando
"join".
Se l'opzione 'cpoptions' contiene anche il flag 'j', farlo solo dopo
un '.'.
Se no, si inserisce solo uno spazio.
NOTA: Quest'opzione è impostata a on se si attiva 'compatible'.

'key'
'key' stringa (default: "")
locale al buffer
{non in Vi}
{disponibile solo se compilato con la funzionalità
|**+cryptv**|}
La chiave da usare per cifrare e decifrare il buffer corrente.
Vedere |**encryption**| e 'cryptmethod'.
Attenzione: Non impostare il valore della chiave a mano, qualcuno
potrebbe vedere la chiave che state inserendo. Usate il comando
|**:X**| . Ma potete annullare il valore di 'key': >
:set key=
< Non è possibile visualizzare il valore di quest'opzione usando
":set key" o "echo &key". Il motivo è quello di evitare di mostrarla
a qualcuno che non dovrebbe conoscerla. Questo implica anche che voi
stessi non potete visualizzarla una volta che l'avete impostata,
quindi state attenti a non fare errori di battitura!
È possibile usare "&key" in un'espressione, per determinare se la

cifratura è abilitata. Quando 'key' è impostata, restituisce come valore "*****" (cinque asterischi).

'keymap' *'kmp'* *E544*

'keymap' 'kmp' stringa (default: "")
locale al buffer
{non in Vi}
{disponibile solo se compilato con la funzionalità |+keymap|}

Nome di una mappatura di tastiera [keymap]. Vedere |mbyte-keymap|. L'impostazione di quest'opzione a un nome valido di keymap ha come effetto secondario di impostare 'iminsert' a uno, in modo che la keymap divenga effettiva.
'imsearch' è pure impostato a uno, a meno che valga -1.
Solo caratteri normali possono essere usati nei nomi di file, i caratteri "/*?[|<>" non sono validi.

'keymodel' *'km'*

'keymodel' 'km' stringa (default: "")
globale
{non in Vi}

Lista di parole separate da virgola che abilitano delle azioni speciali che i tasti sono in grado di svolgere. Questi valori si possono usare:

startsel	Usando un tasto speciale in maiuscolo, si inizia una selezione (in modo Select o modo Visual, a seconda del "tasto" indicato in 'selectmode').
stopsel	Usando un tasto speciale non in maiuscolo, termina la selezione.

I tasti speciali in questo contesto sono quelli del cursore, <End>, <Home>, <PageUp> e <PageDown>.
L'opzione 'keymodel' è impostata dal comando |**:behave**|.

'keywordprg' *'kp'*

'keywordprg' 'kp' stringa (default: "man" o "man -s", DOS: ":help", VMS: "help")
globale o locale al buffer |**global-local**|
{non in Vi}

Programma da usare per il comando |**K**|. Le variabili d'ambiente vengono valutate |**:set_env**|. ":help" si può usare, per accedere all'help interno di Vim. (Nota Nel passato, impostando l'opzione globale col valore della stringa nulla faceva quest'ultima cosa, che adesso è sconsigliata.)
Quando il primo carattere è ":", il comando è chiamato come un comando Ex di Vim, premettendogli l'eventuale [count].
Quando si usa "man", "man -s" o un comando Ec, Vim in automatico traduce l'eventuale contatore fornito assieme al comando "K" e lo passa al comando come primo argomento. Per "man -s", il "-s" viene tolto se non viene indicato un contatore.
Vedere |**option-backslash**| per inserire spazi e backslash.
Esempio: >

```
:set keywordprg=man\ -s
```

< Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'langmap' *'lmap'* *E357* *E358*

'langmap' 'lmap' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |+langmap|}

Quest'opzione consente di passare la vostra tastiera in modo da immettere caratteri in una lingua speciale, differente da quella della tastiera. Se state battendo del testo in modo Insert i caratteri sono immediatamente inseriti. Quando siete in modo Normal l'opzione 'langmap' si cura di ritradurre questi caratteri speciali al significato originale del tasto. Questo significa che non si deve cambiare la modalità della tastiera per poter eseguire dei comandi in modo Normal.
Questa è l'opposto dell'opzione 'keymap', nella quale i caratteri sono mappati in modo Insert.
Si valuti se disattivare 'langoremap' per evitare che 'langmap' sia

applicato anche a caratteri provenienti da una mappatura.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

Esempio (per il greco, in UTF-8): *greek* >
```
:set langmap=Î¿A,Î¿B,Î¨C,Î¿D,Î¿E,Î¦F,Î¿G,Î¿H,Î¿I,Î¿J,Î¿K,Î¿L,Î¿M,Î¿N,Î¿O
,Î P,QQ,Î¡R,Î£S,Î¤T,Î¿U,Î©V,WW,Î§X,Î¥Y,Î¿Z,Î±a,Î²b,Ï¿c,Î´d,Îµe,Ï¿f,Î³g,Î·h,Î¹i,Î¾j,Î
ºk,Î»l,Î¼m,Î½n,Î¿o,Ï¿p,qq,Ï¿r,Ï¿s,Ï¿t,Î¸u,Ï¿v,Ï¿w,Ï¿x,Ï¿y,Î¶z
```
<	Esempio (scambia fra loro il significato di z e y nei comandi): >
```
:set langmap=zy,yz,ZY,YZ
```
<
L'opzione 'langmap' è una lista di parti, separate da virgola. Ogni parte può essere in una delle seguenti due forme:
1. Una lista di coppie. Ogni coppia è un carattere "originale", immediatamente seguito dal carattere di "sostituzione". Ad es.: "aA", "aAbBcC".
2. Una lista di caratteri "originali", un punto e virgola, e una lista di caratteri di "sostituzione". Ad es.: "abc;ABC"

Esempio: "aA,fgh;FGH,cCdDeE"
I caratteri speciali devono essere protetti con un backslash. Questi caratteri sono: ";", ',' e il backslash [\] stesso.

Questo vi consente di attivare azioni di vim senza dover continuare a cambiare la lingua della tastiera. I caratteri della vostra lingua verranno interpretati come normali caratteri inglesi di vim (in conformità con le mappature 'langmap') nei casi seguenti:
- modo Normal/Visual (comandi, nomi di buffer/registri, mappature utente)
- modo Insert/Replace: Nomi di registri dopo aver battuto CTRL-R
- modo Insert/Replace: Mappature

I caratteri immessi in modo Command-line NON saranno toccati da questa opzione. Nota Quest'opzione può essere cambiata in ogni momento, consentendo di alternare mappature per differenti lingue/codifiche. Usate una mappatura, per evitare di dover riscrivere quest'opzione ogni volta.

'langmenu' *'lm'*
'langmenu' 'lm' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con le funzionalità |**+menu**| e |**+multi_lang**|}

Lingua da usare per tradurre il menù. Specifica quale file sarà caricato dalla directory "lang" in 'runtimepath': >
```
"lang/menu_" . &langmenu . ".vim"
```
<	(senza spazi). Ad es., per usare sempre i menù olandesi, a prescindere dal valore a cui è impostato $LANG: >
```
:set langmenu=nl_NL.ISO_8859-1
```
<	Se 'langmenu' non è impostato, si usa |**v:lang**|.
Solo caratteri normali possono essere usati nei nomi di file, i caratteri "/*?[|<>" non sono validi.
Se il vostro $LANG è impostato a una lingua diversa dall'inglese, ma volete usare i menù in inglese: >
```
:set langmenu=none
```
<	Quest'opzione va impostata prima di caricare i menù, di abilitare la scoperta dei tipi di file, o l'evidenziazione sintattica.
Una volta che i menù siano stati definiti, l'impostare quest'opzione non produce alcun effetto. Ma si potrebbe fare questo: >
```
:source $VIMRUNTIME/delmenu.vim
:set langmenu=de_DE.ISO_8859-1
:source $VIMRUNTIME/menu.vim
```
<	Attenzione: Questo annulla tutti i menù definiti da voi!

'langnoremap' *'lnr'* *'nolangnoremap'* *'nolnr'*
'langnoremap' 'lnr' booleana (default: off, impostata a on in |**defaults.vim**|)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+langmap**|}

Quest'opzione è equivalente a 'langremap', ma a valori invertiti.
È disponibile solo per assicurare compatibilità all'indietro.

Se si imposta l'opzione 'langremap', 'langnoremap' viene impostata
al valore opposto, e viceversa.

'langremap' *'lrm'* *'nolangremap'* *'nolrm'*
'langremap' 'lrm' booleana (default: on, messa a off in |**defaults.vim**|)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|**+langmap**|}
Se vale off, l'opzione 'langmap' non si applica ai caratteri prodotti
tramite una mappatura. Questo significa sostanzialmente che se si
nota, impostando 'langmap', che alcune delle mappature risultano
disabilitate,si può provare a impostare quest'opzione. Questa
opzione è attiva per default per compatibilità all'indietro.
Va messa a off se può servire a evitare che delle mappature
cessino di funzionare.

'laststatus' *'ls'*
'laststatus' 'ls' numero (default: 1)
globale
{non in Vi}
Il valore di quest'opzione determina quando la finestra più in basso
avrà una riga di status:
0: mai
1: solo se ci sono almeno due finestre
2: sempre
Lo schermo ha un aspetto migliore con una riga di status, se avete
molte finestre, ma la riga utilizza un'altra riga dello schermo.
|**status-line**|

'lazyredraw' *'lz'* *'nolazyredraw'* *'nolz'*
'lazyredraw' 'lz' booleana (default: off)
globale
{non in Vi}
Quando quest'opzione è impostata, lo schermo non verrà aggiornato
durante l'esecuzione di macro, registri e altri comandi che non siano
stati battuti direttamente. Inoltre, l'aggiornamento del titolo della
finestra è differito. Per farlo eseguire subito, usare |**:redraw**|.

'linebreak' *'lbr'* *'nolinebreak'* *'nolbr'*
'linebreak' 'lbr' booleana (default: off)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+linebreak**|}
Se impostato a on Vim spezzerà delle righe che non stanno in una riga
di schermo quando incontro un carattere specificato in 'breakat'
invece che all'ultimo carattere che può essere contenuto nella riga
dello schermo. A differenza di 'wrapmargin' e 'textwidth', questo
non causa l'aggiunta di caratteri di fine riga <EOL> nel file, ma
riguarda solo il modo in cui il file è visualizzato, non i suoi
contenuti. Il valore di 'showbreak' è usato per marcare l'inizio delle
righe visualizzate nelle righe successivo dello schermo.
quest'opzione non è utilizzata quando l'opzione 'wrap' è impostata a
off.
Nota I caratteri <Tab> dopo un carattere di fine riga <EOL> sono per
lo più visualizzati con un numero di spazi bianchi non corretto.

'lines' *E593*
'lines' numero (default: 24 o altezza del terminale)
globale
Numero di righe della finestra Vim.
Di solito non è necessario impostarlo. L'impostazione è fatta
automaticamente dal programma che inizializza il terminale.
Vedere anche |**posix-screen-size**|.
Se Vim viene eseguito nella GUI o in una finestra che
può essere ridimensionata, l'impostazione di quest'opzione
modificherà la dimensione della finestra. Se si vuole usare questa
opzione solo per la GUI, mettere il vostro comando nel file |**gvimrc**|.
Vim limita il numero di righe a quelle che lo schermo può contenere.
Potete usare il comando seguente per ottenere la finestra più alta
possibile: >
:set lines=999

< Il valore minimo è 2, quello massimo è 1000.
Se si ottengono meno righe di quel che si aspettava, si controlli l'opzione 'guiheadroom'.
Quando impostate quest'opzione e Vim non è in grado di cambiare il numero delle righe dello schermo, lo schermo può essere visualizzato in maniera incorretta.

'linespace' *'lsp'*
'linespace' 'lsp' numero (default: 0, 1 per Win32 GUI)
globale
{non in Vi}
{solo nella GUI}
Numero di righe di pixel da inserire fra i caratteri. Utile se il font usa per intero l'altezza del rettangolo che contiene il carattere, facendo sì che le righe siano a contatto fra loro. Se diverso da zero, c'è spazio per la sottolineatura.
Con alcuni font ci può essere troppo spazio fra le righe (per aver spazio per dei caratteri che "scendono" o "salgono"). In questo caso ha senso impostare 'linespace' a un valore negativo. Questo potrebbe però creare problemi a livello di schermo!

'lisp' *'nolisp'*
'lisp' booleana (default: off)
locale al buffer
{non disponibile se compilato senza la funzionalità |**+lispindent**|}
Modo Lisp: Quando si batte <Invio> in modo Insert imposta l'indentatura della riga seguente secondo gli standard Lisp (beh, all'incirca). Questo succede anche con "cc" o "S".
Anche 'autoindent' deve essere stato specificato, perché questo accada. Il flag 'p' di 'cpoptions' modifica la modalità di indentatura: compatibile con Vi, o migliorato. Vedere anche 'lispwords'.
Il carattere '-' è incluso fra i caratteri chiave. Ridefinisce l'operatore "=" per usare questo stesso algoritmo di indentatura invece che chiamare un programma esterno, se l'opzione 'equalprg' è nulla.
Quest'opzione non è utilizzata quando l'opzione 'paste' è impostata.
{Vi: Si comporta in maniera leggermente differente}

'lispwords' *'lw'*
'lispwords' 'lw' stringa (il default è molto lungo)
globale o locale al buffer |**global-local**|
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+lispindent**|}
Lista di parole, separate da virgola, che influenzano l'indentatura Lisp. |**'lisp'**|

'list' *'nolist'*
'list' booleana (default: off)
locale alla finestra
Modo List: Visualizza tabulazioni come CTRL-I, inserisce "$" dopo la fine riga. Utile per visualizzare la differenza fra tabulazioni e spazi e per gli spazi bianchi a fine riga. La visualizzazione può essere ulteriormente modificato dall'opzione 'listchars'.

Il cursore è visualizzato all'inizio dello spazio che il carattere <Tab> occupa, e non alla fine, come solitamente in modo Normal. Per ottenere la posizione del cursore come in modo Normal, mentre si visualizzano le <Tab> con gli spazi, usare: >
:set list lcs=tab:\ \
<
Nota Questo influenzerà anche la formattazione (impostata con 'textwidth' o 'wrapmargin') se 'cpoptions' comprende 'L'. Vedere 'listchars' per cambiare il modo con cui le tabulazioni sono visualizzate.

'listchars' *'lcs'*
'listchars' 'lcs' stringa (default: "eol:$")
globale
{non in Vi}
Stringhe di caratteri da usare in modo 'list' e per il comando

|:list|. È una lista di impostazioni di stringhe separate da virgola.
lcs-eol
eol:c Carattere da mostrare a fine di ogni riga. Se omesso, non c'è nessun carattere in più a fine riga.
lcs-tab
tab:xy Due caratteri usati per mostrare <Tab>. Il primo carattere è usato una volta sola. Il secondo carattere è ripetuto quanto basta a riempire lo spazio normalmente occupato da <Tab>.
"tab:>-" mostrerà una <Tab> che riempie quattro spazi come ">---". Se omesso, <Tab> è visualizzata come ^I.
lcs-space
space:c Carattere da mostrare for uno spazio. Se omesso, gli spazi sono lasciati bianchi.
lcs-trail
trail:c Carattere da mostrare per spazi in più a fine riga. Se omesso, gli spazi in più restano non evidenziati. Prevale sull'impostazione "space" per il caso di spazi a fine riga.
lcs-extends
extends:c Carattere da mostrare nell'ultima colonna dello schermo quando 'wrap' è a off e la riga continua oltre il lato destro dello schermo.
lcs-precedes
precedes:c Carattere da mostrare nella prima colonna dello schermo quando 'wrap' è a off e c'è del testo che precede il primo carattere visibile nella prima colonna dello schermo.
lcs-conceal
conceal:c Character da mostrare al posto del testo nascosto, quando 'conceallevel' vale 1.
lcs-nbsp
nbsp:c Carattere da mostrare per il carattere spazio bianco non interrompibile [dopo il quale non si può andare a capo -NdT] (carattere 0xA0, 160). Lasciato bianco se omesso.

I caratteri ':' e ',' non andrebbero usati. Caratteri UTF-8 possono essere specificati quando l''encoding' è "utf-8", diversamente, sono permessi solo caratteri stampabili. Tutti i caratteri devono essere di larghezza singola.

Esempi: >
:set lcs=tab:>-,trail:-
:set lcs=tab:>-,eol:<,nbsp:%
:set lcs=extends:>,precedes:<
< L'evidenziazione "NonText" sarà usata per "eol", "extends" e "precedes". Quella "SpecialKey" per "nbsp", "space", "tab" e "trail". |**hl-NonText**| |**hl-SpecialKey**|

'lpl' *'nolpl'* *'loadplugins'* *'noloadplugins'*
'loadplugins' 'lpl' booleana (default: on)
globale
{non in Vi}
Se impostata gli script plugin sono caricati a inizio sessione |**load-plugins**|. Quest'opzione può essere impostata a off nel vostro file |**vimrc**| per non effettuare il caricamento di plugin.
Nota Quando si usano gli argomenti "-u NONE", "-u DEFAULTS" e "--noplugin" nella riga di comando, quest'opzione viene impostata a off.
Vedere |**-u**| e |**--noplugin**|.

'luadll'
'luadll' stringa (default: dipende dalla compilazione di Vim)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+lua/dyn**|}
Specifica il nome della libreria condivisa Lua. Il default è DYNAMIC_LUA_DLL, specificato al momento della compilazione.
Le variabili d'ambiente sono valutate |**:set_env**|.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'macatsui' *'nomacatsui'*
'macatsui' booleana (default: on)
globale
{disponibile solo nella versione GUI Mac}
Questo è un tentativo di rimediare quando la videata non è mostrata correttamente. Quando Vim è impostato con il support multi-byte, si usa la visualizzazione di testo ATSUI. Quando non impostata, non si usa la visualizzazione di testo ATSUI. Inattivate quest'opzione se avete problemi di visualizzazione, In una futura versione di Vim i problemi potrebbero essere stati risolti e quest'opzione diverrebbe obsoleta. Per questo, usate il metodo seguente per disattivarla: >

```
if exists('&macatsui')
  set nomacatsui
endif
```

< Un'altra opzione da controllare se avete problemi per ridisegnare lo schermo è 'termencoding'.

'magic' *'nomagic'*
'magic' booleana (default: on)
globale
Cambia i caratteri speciali utilizzabili nei modelli di ricerca (espressioni regolari).
Vedere |**pattern**|.
ATTENZIONE: Se quest'opzione è a off, molti plugin potrebbero non funzionare! In effetti molte espressioni di ricerca presuppongono che l'opzione sia a on, e non funzioneranno quando essa è a off. Va impostata a off solo si utilizzano vecchi script Vi. In ogni altra situazione vanno usati dei modelli (espressioni di ricerca) che funzionino quando 'magic' è impostato a on.
Includere "\M" se si vuole |**/\M**|.

'makeef' *'mef'*
'makeef' 'mef' stringa (default: "")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+quickfix**|}
Nome del file degli errori per il comando |**:make**| (vedere |**:make_makeprg**|) e il comando |**:grep**|.
Quando non è impostato, un file temporaneo generato internamente verrà utilizzato.
Quando "##" è incluso, sarà rimpiazzato da un numero per rendere unico il nome del file. Questo dà la certezza che il comando ":make" non andrà a ricoprire un file già esistente.
NON usato per il comando ":cf". Vedere 'errorfile' per questo caso.
Le variabili d'ambiente vengono valutate |**:set_env**|.
Vedere |**option-backslash**| per inserire spazi e backslash.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'makeencoding' *'menc'*
'makeencoding' 'menc' stringa (default "")
globale o locale al buffer |**global-local**|
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+multi_byte**|}
Codifica usata per leggere l'output prodotto da comandi esterni. Se il valore è la stringa nulla, la codifica non è convertita.
Opzione usata da `:make`, `:lmake`, `:grep`, `:lgrep`, `:grepadd`, `:lgrepadd`, `:cfile`, `:cgetfile`, `:caddfile`, `:lfile`, `:lgetfile` e `:laddfile`.

Quest'opzione è utile principalmente se si sta usando MS-Windows e l'opzione 'encoding' vale "utf-8". Se |**+iconv**| è attivato, e si sta usando la libreria GNU libiconv, impostare 'makeencoding' a "char" ha lo stesso effetto che impostare la codifica a quella locale.
Esempio: >

```
:set encoding=utf-8
:set makeencoding=char  " si usa la localizzazione di sistema
```

<

'makeprg' *'mp'*
'makeprg' 'mp' stringa (default: "make", VMS: "MMS")
globale o locale al buffer |**global-local**|

{non in Vi}
Programma da usare per il comando ":make". Vedere |**:make_makeprg**|.
Quest'opzione può contenere caratteri '%' e '#', (vedere |**:_%**| |**:_#**|),
che sono valutati come nome del file corrente, e come nome alternativo
del file. Si usi |**::S**| per proteggere i nomi di file, nel caso in cui
contengano caratteri speciali.
Le variabili d'ambiente vengono valutate |**:set_env**|. Vedere
|**option-backslash**| per inserire spazi e backslash.
Nota Un carattere '|' deve essere protetto due volte: una volta per il
comando ":set" e una seconda volta per l'interpretazione di un
comando. Per usate un filtro chiamato "myfilter", scrivete così: >
:set makeprg=gmake\ \\\|\ myfilter
< Il segnaposto "$*" può essere incluso (anche più volte), per specificare
dove vanno inseriti gli argomenti, ad es.: >
:set makeprg=latex\ \\\\nonstopmode\ \\\\input\\{$*}
< Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|,
per motivi di sicurezza.

'matchpairs' 'mps' stringa (default: "(:),{:},[:]")
locale al buffer
{non in Vi}
Caratteri che formano coppie. Il comando |**%**| salta dall'uno
all'altro.
Per ora solo coppie di caratteri singoli sono consentite, e questi
devono essere differenti fra loro, ragion per cui non si può saltare
tra una coppia di doppi apici.
I caratteri devono essere separati da un ":".
Le coppie devono essere separate da una virgola.
Esempio per includere '<' e '>' (HTML): >
:set mps+=<:>

< Esempio più esotico, per saltare tra '=' e ';' in una proposizione di
assegnazione, utile per linguaggi come C e Java: >
:au FileType c,cpp,java set mps+==;

< Per una maniera più avanzata di utilizzare "%", vedere il plugin
matchit.vim nella directory $VIMRUNTIME/pack/dist/opt/matchit.
|**add-local-help**|

'matchtime' *'mat'*
'matchtime' 'mat' numero (default: 5)
globale
{non in Vi}{in Nvi}
Decimi di secondo durante i quali mostrare la parentesi
corrispondente, se 'showmatch' è impostato. Nota Questo tempo non è
espresso in millisecondi, come altre opzioni che impostano una durata.
Questo è stato fatto per mantenere la compatibilità con Nvi.

'maxcombine' *'mco'*
'maxcombine' 'mco' numero (default: 2)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|**+multi_byte**|}
Il massimo numero di caratteri uniti fra loro supportato per la
visualizzazione.
Usata solo se 'encoding' è "utf-8".
Il default va bene per la laggioranza delle lingue. L'ebraico può
aver bisogno di 4.
Il valore massimo è 6.
Anche se quest'opzione è impostata a 2, si possono ancora editare
testi contenenti più di 2 caratteri uniti tra loro, ma non sono
visualizzati. Usare |**g8**| o |**ga**|.
Vedere |**mbyte-combining**|.

'maxfuncdepth' *'mfd'*
'maxfuncdepth' 'mfd' numero (default: 100)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+eval**|}
Massima profondità di invocazione di funzione per funzioni utente.
Questo serve di solito per tenere sotto controllo una ricorsività

infinita. Quando si usa una funzione ricorsiva che richieda una maggiore profondità, impostare 'maxfuncdepth' a un numero più elevato. In questo modo, però, si utilizzerà più memoria, correndo il rischio di interrompere l'esecuzione quando la memoria sia esaurita.
Se questo limite supera il valore 200, viene anche modificato il valore massimo per la ricorsività del comando ex, vedere |**E169**|.
Vedere anche |**:function**|.

'maxmapdepth' *'mmd'* *E223*

'maxmapdepth' 'mmd' numero (default: 1000)
globale
{non in Vi}

Numero massimo di volte in cui si può effettuare una mappatura senza che questa risulti in un carattere da usare effettivamente. Questo serve di solito per tenere sotto controllo una mappatura infinita, come ad es. ":map x y" con ":map y x". Non si riesce ancora in questo modo a neutralizzare ":map g wg", perché il 'w' è usato prima di effettuare la mappatura successiva. Vedere anche |**key-mapping**|.

'maxmem' *'mm'*

'maxmem' 'mm' numero (default: tra 256 e 5120 (a seconda del sistema) o metà della quantità di memoria disponibile)
globale
{non in Vi}

Massima quantità di memoria (in Kbyte) da usare per un buffer. Quando questo limite è stato raggiunto, l'allocazione di memoria ulteriore per un buffer causerà il rilascio di altra memoria.
Il valore massimo utilizzabile è circa 2000000. Usare questo valore per lavorare senza un limite.
Questo valore è ignorato se 'swapfile' è impostato a off.
Vedere anche 'maxmemtot'.

'maxmempattern' *'mmp'*

'maxmempattern' 'mmp' numero (default: 1000)
globale
{non in Vi}

Massima quantità di memoria (in Kbyte) da usare per trovare corrispondenze nelle ricerche.
Il valore massimo è circa 2000000. Usare questo valore per lavorare senza imporre un limite.
E363
Una volta che Vim raggiunge il limite specificato, dà un messaggio di errore, e si comporta quasi allo stesso modo che se si fosse immesso CTRL-C.
Raggiungere il limite spesso significa che l'espressione di ricerca è molto inefficiente o troppo complessa. Questo può capitare con l'espressione "\(.\)*" applicata a una riga molto lunga. Specificare ".*" funziona molto meglio.
Vim può esaurire la memoria disponibile anche prima di raggiungere il limite specificato con 'maxmempattern'.

'maxmemtot' *'mmt'*

'maxmemtot' 'mmt' numero (default: tra 2048 e 10240 (a seconda del sistema) o metà della quantità di memoria disponibile)
globale
{non in Vi}

Massima quantità di memoria (in Kbyte) da usare in totale per tutti i buffer. Il valore massimo è circa 2000000 (2 Gbyte). Usare questo valore per lavorare senza un limite.
Su macchine a 64 bit valori più alti potrebbero essere usati. Ma, ci si domandi, servono davvero più di 2 Gbyte per editare dei testi? Si tenga presente che il testo è immagazzinato nel file di swap, e che quindi si possono editare in ogni caso file più grandi di 2 Gbyte. La memoria è necessaria per memorizzare le informazioni che rendono possibile il comando undo.
I buffer per i quali l'opzione 'swapfile' vale off contribuiscono comunque a determinare il totale della memoria usata.
Vedere anche 'maxmem'.

'menuitems' *'mis'*

'menuitems' 'mis' numero (default: 25)

globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+menu**|}
Massimo numero di elementi da usare in un menù. Usato per menù che sono generati da una lista di elementi, ad es., il menù dei Buffer. Il cambiamento di quest'opzione non ha un effetto immediato, occorre prima rinfrescare il menù.

'mkspellmem' *'msm'*
'mkspellmem' 'msm' stringa (default: "460000,2000,500")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+syntax**|}
Parametri per |**:mkspell**|. Essi consentono di decidere quando iniziare a comprimere l'albero delle parole [ossia la lista di parole "valide" per una certa lingua. La compressione può essere lenta quando ci sono molte parole, ma è necessaria per evitare di esaurire la memoria disponibile. La quantità di memoria utilizzata per ogni parola dipende principalmente da quanto sono simili fra loro le parole, e questa è la ragione per cui questa regolazione è complessa.

Ci sono tre numeri, separati da virgola:
{iniziale},{incremento},{aggiunta}

Per molte lingue l'albero di parole non compresso può essere interamente contenuto in memoria. {iniziale} specifica la quantità di memoria in Kbyte che si può usare prima di effettuare una compressione. Dovrebbe essere un filo minore della quantità massima di memoria a disposizione di Vim.

Quando si supera il limite {iniziale} il numero {incremento} specifica la quantità di memoria in Kbyte che può essere allocata, prima di effettuare un'altra compressione. Un numero basso significa che la compressione viene effettuata dopo aver aggiunto un numero minore di parole, il che rallenta la procedura. Un numero alto significa che verrà allocata una maggior quantità di memoria.

Dopo aver effettuato una compressione, un numero di parole uguale ad {aggiunta} moltiplicato per 1024 words può essere aggiunto, prima di ignorare il limite {incremento} e si fa la compressione quando serve una ulteriore quantità di memoria. Un numero basso significa che c'è una minore probabilità di raggiungere il limite {incremento}, che si usa meno memoria, ma che la procedura è più lenta.

Le lingue per le quali questi numeri sono importanti sono l'italiano e l'ungherese. Il default funziona nel caso in cui disponiate di 512Mbyte. Se avete a disposizione 1 Gbyte, potreste usare: >
:set mkspellmem=900000,3000,800
< Se avete a disposizione meno di 512 Mbyte |**:mkspell**| può interrompersi per alcune lingue, a prescindere dai valori specificati in 'mkspellmem'.

'modeline' *'ml'* *'nomodeline'* *'noml'*
'modeline' 'ml' booleana (default Vim: on, (off per "root"),
default Vi: off)
locale al buffer
'modelines' *'mls'*
'modelines' 'mls' numero (default: 5)
globale
{non in Vi}
Se 'modeline' è a on, 'modelines' dà il numero di righe controllate alla ricerca di comandi "set". Se 'modeline' è a off o 'modelines' è a zero nessuna riga viene controllata. Vedere |**modeline**|.
NOTA: 'modeline' è impostata al valore di default di Vi impostando 'compatible' e al valore di default di Vim quando 'compatible' viene messa a off.

'modifiable' *'ma'* *'nomodifiable'* *'noma'*
'modifiable' 'ma' booleana (default: on)
locale al buffer
{non in Vi} *E21*

Se a off il contenuto del buffer non può essere modificato. Neppure le opzioni 'fileformat' e 'fileencoding' possono essere cambiate. Può essere annullato quando si richiama Vim (rendendo impossibili modifiche) con l'argomento |**-M**| della riga di comando.

'modified' *'mod'* *'nomodified'* *'nomod'*

'modified' 'mod' booleana (default: off)
locale al buffer
{non in Vi}

Se impostata, il buffer è considerato come modificato. Quest'opzione è impostata quando:
1. Una modifica è stata apportata al testo dopo l'ultima volta che è stato scritto. Se si usa il comando |**undo**| per ritornare al testo originale (quello che era stato salvato) l'opzione verrà messa a off. Ma se si torna indietro a modifiche effettuate prima che il buffer sia stato salvato l'opzione tornerà di nuovo a on, poiché il testo è diverso da quello che è stato salvato.
2. 'fileformat' o 'fileencoding' è diverso dal suo valore originale. Il valore originale è impostato quando il buffer è letto o scritto. Un comando ":set nomodified" imposterà come originali i valori correnti (di 'fileformat' o 'fileencoding') e l'opzione 'modified' verrà impostata a off.
   Lo stesso vale per 'eol' e 'bomb'.

Quest'opzione non è impostata quando il buffer risulta modificato in seguito ad eventi che innescano gli autocomandi seguenti: BufNewFile, BufRead/BufReadPost, BufWritePost, FileAppendPost o VimLeave.
Vedere |**gzip-example**| per una spiegazione.
Se 'buftype' è "nowrite" o "nofile" quest'opzione può essere impostata, ma sarà ignorata.
Si noti che il testo può in realtà essere lo stesso; p.es. il flag 'modified' viene inpostato se si usa "rA" sul carattere "A".

'more' *'nomore'*

'more' booleana (default Vim: on, default Vi: off)
globale
{non in Vi}

Se impostata, la visualizzazione fa una pausa quando si è riempita l'intera schermata. Se quest'opzione è a off non ci sono pause, la visualizzazione continua fino alla fine.
NOTA: Quest'opzione è impostata al valore di default di Vi impostando 'compatible' e al valore di default di Vim quando 'compatible' viene messa a off.

'mouse' *E538*

'mouse' stringa (default: "", "a" per GUI, MS-DOS e Win32 impostata ad "a" in |**defaults.vim**|)
globale
{non in Vi}

Abilita l'uso del mouse. Disponibile solo per alcuni terminali (xterm, MS-DOS, Win32 |**win32-mouse**|, QNX pterm, *BSD console con sysmouse e Linux console con gpm). Per usare il mouse nella GUI, vedere |**gui-mouse**|.
Il mouse può essere abilitato per modi differenti:

n	modo Normal e modo Terminal
v	modo Visual
i	modo Insert
c	modo Command-line
h	tutti i modi precedenti, se si edita un help-file
a	tutti i modi precedenti
r	per richieste di |**hit-enter**| e |**more-prompt**|

Normalmente si abilita il mouse in tutti e quattro i modi con: >
```
:set mouse=a
```
< Quando il mouse non è abilitato, la GUI userà ugualmente il mouse per una selezione modeless (senza una modalità). Questo non sposta il cursore del testo.

Vedere |**mouse-using**|. Vedere anche |**'clipboard'**|.

Nota: Quando si abilita il mouse in un terminale, copia/incolla usa il registro "* se c'è accesso a un X-server. La modalità xterm dei bottoni del mouse può ancora essere usata tenendo schiacciata il tasto "Maiuscolo" (Shift). Vedere anche l'opzione 'clipboard'.

'mousefocus' *'mousef'* *'nomousefocus'* *'nomousef'*
'mousefocus' 'mousef' booleana (default: off)
globale
{non in Vi}
{funziona solo nella GUI}
La finestra in cui si trova il puntatore del mouse è attivata automaticamente. Se si cambia il formato della finestra o la selezione della finestra in un altro modo, il puntatore del mouse si sposta alla finestra che prende input dalla tastiera. Per default è inattiva, perché rende un po' macchinoso usare i menù a tendina, in quanto un passaggio del puntatore può attivare accidentalmente una finestra.

'mousehide' *'mh'* *'nomousehide'* *'nomh'*
'mousehide' 'mh' booleana (default: on)
globale
{non in Vi}
{funziona solo nella GUI}
Se impostata, il puntatore del mouse è invisibile mentre si immettono dei caratteri. Il puntatore del mouse torna visibile quando il mouse viene spostato.

'mousemodel' *'mousem'*
'mousemodel' 'mousem' stringa (default: "extend", "popup" per MS-DOS e Win32)
globale
{non in Vi}
Imposta il modello da usare per il mouse. Il nome indica principalmente come si vuol usare il pulsante destro del mouse:

extend	Il pulsante destro del mouse estende una selezione. Questo comportamento è simile a quello di un xterm.
popup	Il pulsante destro del mouse fa apparire un menù. Il pulsante sinistro del mouse (tenendo schiacciato il tasto Maiuscolo) estende una selezione. Questo comportamento è simile a quello di Microsoft Windows.
popup_setpos	Come "popup", ma il cursore verrà spostato sulla posizione nella quale il mouse è stato cliccato, e quindi l'operazione selezionata avrà effetto sull'oggetto che è stato cliccato. Se si clicca all'interno di una selezione, quella selezione sarà eseguita, ossia nessuno spostamento del cursore. Questo implica, naturalmente, che il premere il pulsante di destra al di fuori di una selezione farà terminare il modo Visual.

Panoramica di quel che fa il pulsante in ognuno dei modelli:

mouse	extend	popup(_setpos) ~
clic sinistro	posiziona cursore	posiziona cursore
trascinam. sx.	inizia selezione	inizia selezione
Maiusc. sx	cerca parola	estende selezione
clic destro	estende selezione	popup menù (posiziona cursore)
trascinam. dx.	estende selezione	-
clic mediano	incolla	incolla

Nel modello "popup" il pulsante destro del mouse fa apparire un menù. Occorre che il menù sia stato definito in precedenza, vedere |**popup-menu**|.
In un terminale il menu popup funziona se Vim è stato compilato con la funzionalità |**+insert_expand**|.

Nota si può ulteriormente ridefinire il significato dei pulsanti con delle mappature. Vedere |**gui-mouse-mapping**|. Ma le mappature NON sono usate durante una selezione senza modalità (in quanto questa è gestita direttamente dal codice della GUI).

L'opzione 'mousemodel' è impostata dal comando |**:behave**|.

'mouseshape' *'mouses'* *E547*
'mouseshape' 'mouses' stringa (default: "i:beam,r:beam,s:updown,sd:cross,
m:no,ml:up-arrow,v:rightup-arrow")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+mouseshape**|}
Quest'opzione specifica a Vim la forma del puntatore del mouse per

le differenti modalità di Vim. L'opzione è una lista di parti, separata da virgola, simile a quella usata per 'guicursor'. Ogni parte consiste di una coppia costituita da una lista-modi/posizioni e da una lista-argomento:

```
        lista-modi:forma,lista-modi:forma,..
```

La lista-modi è un elenco separato da ":" di questi modi/posizioni:

In una finestra normale: ~

n	modo Normal
v	modo Visual
ve	modo Visual con 'selection' "exclusive" (come 'v', se essa non è specificata)
o	modo Operator-pending
i	modo Insert
r	modo Replace

Altri: ~

c	aggiungendo in fondo alla riga-comandi
ci	inserendo nella riga-comandi
cr	sovrapponendosi nella riga-comandi
m	alla richiesta 'Premi INVIO' o 'Ancora'
ml	idem, ma il cursore è nell'ultima riga
e	ogni modo, puntatore sotto l'ultima finestra
s	ogni modo, puntatore su una riga di stato
sd	ogni modo, trascinando una riga di status
vs	ogni modo, puntatore su una riga di separazione verticale
vd	ogni modo, trascinando una riga di separazione verticale
a	dappertutto

La forma è una delle seguenti:

possib.	nome	aspetto ~
w x	arrow	normale puntatore di mouse (freccia)
w x	blank	nessun puntatore (usare con cautela!)
w x	beam	I-beam (barretta verticale)
w x	updown	freccia verticale a 2 punte per ridimensionare
w x	leftright	freccia orizzont. a 2 punte per ridimensionare
w x	busy	il normale indicatore di sistema occupato
w x	no	il normale puntatore di sistema per 'no input'
x	udsizing	indica ridimensionamento sopra/sotto
x	lrsizing	indica ridimensionamento sinistra/destra
x	crosshair	come un grosso ma sottile segno +
x	hand1	mano nera
x	hand2	mano bianca
x	pencil	ciò con cui si scrive (penna)
x	question	grosso ?
x	rightup-arrow	freccia che punta in alto a destra
w x	up-arrow	freccia che punta in su
x	<number>	un numero di puntatore X11 (vedere X11/cursorfont.h)

la colonna "possib." contiene una 'w' se la forma è disponibile per Win32, x se disponibile per X11.
In ogni modo non indicato o nel caso di forme non disponibili, verrà usato il puntatore di mouse di forma normale.

Esempio: >

```
        :set mouseshape=s:udsizing,m:no
```

< farà sì che il puntatore del mouse si trasformi in una freccia di 'no input' quando viene visualizzato il messaggio 'Premi INVIO' (perché in questa situazione cliccare il mouse non produce alcun effetto).

'mousetime' *'mouset'*

'mousetime' 'mouset' numero (default: 500)
globale
{non in Vi}

Solo per GUI, MS-DOS, Win32 e Unix con xterm. Definisce il tempo massimo, in millisecondi, tra due cliccate di mouse, per far riconoscere il secondo clic come un multi-clic.

'mzschemedll' stringa (default: dipende dalla compilazione di Vim)
globale

{non in Vi}
{disponibile solo se compilato con la funzionalità
|+mzscheme/dyn|}
Specifica il nome della libreria condivisa MzScheme. Il default è DYNAMIC_MZSCH_DLL, specificato al momento della compilazione.
Il valore va impostato nello script |vimrc| o ancora prima. In fase di avvio, prima del passo di caricamento dei plugin (load-plugins).
Le variabili d'ambiente sono valutate |:set_env|.
Quest'opzione non può essere impostata da |modeline|, o nel |sandbox|, per motivi di sicurezza.

'mzschemegcdll' stringa (default: dipende dalla compilazione di Vim)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+mzscheme/dyn|}
Specifica il nome della libreria condivisa MzScheme GC. Il default è DYNAMIC_MZGC_DLL, specificato al momento della compilazione.
Il valore può essere uguale a quello di 'mzschemedll' se il codice GC è ivi incluso.
Le variabili d'ambiente sono valutate |:set_env|.
Quest'opzione non può essere impostata da |modeline|, o nel |sandbox|, per motivi di sicurezza.

'mzquantum' *'mzq'*
'mzquantum' 'mzq' numero (default: 100)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|+mzscheme|}
Il numero di millisecondi fra un controllo e l'altro dei "thread" MzScheme (i "thread" sono dei lavori che procedono in parallelo fra loro, e quindi richiedono una sincronizzazione).
Valori minori o uguali a zero indicano che non verranno schedulati del "thread".
NOTA: Quest'opzione è impostata al valore di default di Vim quando si imposta a off 'compatible'.

'nrformats' *'nf'*
'nrformats' 'nf' stringa (default: "bin,octal,hex",
impostata a "bin,hex" in |defaults.vim|)
locale al buffer
{non in Vi}
Quest'opzione definisce quale sarà la base da utilizzare per i numeri quando si usano i comandi CTRL-A e CTRL-X per aggiungere e sottrarre, rispettivamente, da un numero; vedere |CTRL-A| per ulteriori informazioni su questi comandi.
alpha Se specificato, singoli caratteri alfabetici saranno incrementati o decrementati. Ciò può tornare utile per una lista con un indice letterale a), b), etc.
octal-nrformats
octal Se specificato, numeri che iniziano per zero saranno considerati essere ottali. Ad es.: Usando CTRL-A su "007" produrrà "010".
hex Se specificato, numeri che iniziano con "0x" o "0X" saranno considerati essere esadecimali. Ad es.: Usando CTRL-X su "0x100" produrrà "0x0ff".
bin Se specificato, numeri che iniziano con "0b" o "0B" saranno considerati essere binari. Ad es:: Usando CTRL-X su "0b1000" sottrae uno, producendo "0b0111".
Numeri che iniziano semplicemente con una cifra compresa tra 1 e 9 sono sempre considerati decimali. Questo succede anche per numeri che non sono riconosciuti essere ottali o esadecimali.

'number' *'nu'* *'nonumber'* *'nonu'*
'number' 'nu' booleana (default: off)
locale alla finestra
Mostra il numero della riga davanti a ogni riga. Quando l'opzione 'n' è assente da 'cpoptions' una riga visualizzata su più righe dello schermo non userà la colonna contenente i numeri di riga (questo è il default quando 'compatible' è messa a off).
L'opzione 'numberwidth' può essere usata per specificare lo spazio da usare per metterci il numero di riga.

Quando una riga lunga, visualizzata su una sola riga dello schermo non inizia con il primo carattere, dei caratteri '-' sono messi prima del numero.
Vedere |**hl-LineNr**| e |**hl-CursorLineNr**| per l'evidenziazione usata per il numero.

number_relativenumber

L'opzione 'relativenumber' rende il numero visualizzato relativo al cursore. Insieme con 'number' sono possibili queste quattro combinazioni (cursore alla riga 3):

```
        'nonu'          'nu'            'nonu'          'nu'
        'nornu'         'nornu'         'rnu'           'rnu'

    |apple          |  1 apple      |  2 apple      |  2 apple
    |pear           |  2 pear       |  1 pear       |  1 pear
    |nobody         |  3 nobody     |  0 nobody     |3   nobody
    |there          |  4 there      |  1 there      |  1 there
```

'numberwidth' *'nuw'*

'numberwidth' 'nuw' numero (default Vim: 4, default Vi: 8)
locale alla finestra
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+linebreak**|}

Numero minimo di colonne da usare per metterci il numero di riga. Serve solo quando l'opzione 'number' o 'relativenumber' siano impostate o se si stanno stampando righe precedute dal numero di riga. Poiché c'è sempre uno spazio fra il numero e il testo, rimane un carattere in meno per il numero vero e proprio.
Questo valore è la larghezza minima. Una larghezza maggiore è usata se ciò è necessario per contenere il numero più alto presente nel buffer, che può essere il numero più alto di righe nel buffer o il numero di righe nella finestra, a seconda se sia impostata l'opzione 'number' oppure 'relativenumber'. Quindi con il default di Vim (4) c'è posto per un numero di riga che arrivi al massimo a 999. Quando il buffer raggiunge le 1000 righe, saranno usate 5 colonne.
Il valore minimo è 1, il massimo è 10.
NOTA: Quest'opzione è impostata al valore di default di Vi quando si attiva 'compatible' e al valore di default di Vim quando si imposta a off 'compatible'.

'omnifunc' *'ofu'*

'omnifunc' 'ofu' stringa (default: "")
locale al buffer
{non in Vi}
{non disponibile se compilato senza le funzionalità |**+eval**| o |**+insert_expand**|}

Quest'opzione specifica una funzione da usare in modo Insert per il completamento "omni" con CTRL-X CTRL-O. |**i_CTRL-X_CTRL-O**|
Vedere |**complete-functions**| per una spiegazione di come la funzione è invocata e di cosa dovrebbe fornire.
Quest'opzione è di solito impostata da un "plugin" per un tipo file: |**:filetype-plugin-on**|
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'opendevice' *'odev'* *'noopendevice'* *'noodev'*

'opendevice' 'odev' booleana (default: off)
globale
{non in Vi}
{solo per MS-DOS, MS-Windows e OS/2}

Abilita lettura da e scrittura su dispositivo (device, ossia direttamente da/su pezzi di hardware). Ciò può causare un blocco di Vim su un dispositivo che può essere aperto, ma che non effettua direttamente le attività di I/O (Input/Output). Per questo l'opzione è inattiva per default.
Nota In ambiente MS-Windows editare i file "aux.h", "lpt1.txt" e simili equivale anche a editare un dispositivo.

'operatorfunc' *'opfunc'*

'operatorfunc' 'opfunc' stringa (default: "")
globale
{non in Vi}

Quest'opzione specifica una funzione da invocare da parte
dell'operatore |**g@**|.
Vedere |**:map-operator**| per informazioni ulteriori e per un esempio.

Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|,
per motivi di sicurezza.

'osfiletype' *'oft'*
'osfiletype' 'oft' stringa (default: "")
locale al buffer
{non in Vi}
{disponibile solo se compilato con la funzionalità
|**+osfiletype**|}
Quest'opzione era supportata per solo RISC OS, ed è stata rimossa.

'packpath' *'pp'*
'packpath' 'pp' string (default: vedere 'runtimepath')
{non in Vi}
Directory usata per trovare pacchetti di plugin. Vedere |**packages**|.

'paragraphs' *'para'*
'paragraphs' 'para' string (default: "IPLPPPQPP TPHPLIPpLpItpplpipbp")
globale
Specifica le macro "nroff" che delimitano i paragrafi. Si tratta di
coppie di due lettere (vedere |**object-motions**|).

'paste' *'nopaste'*
'paste' booleana (default: off)
globale
{non in Vi}
Mette Vim in modo Paste. Questo torna utile se si vuole tagliare o
copiare del testo da una finestra e incollarlo in Vim. Si evitano
così effetti imprevisti.
È utile impostare quest'opzione se si usa Vim in un terminale, dove
Vim non è in grado di distinguere fra testo inserito e testo
incollato. Nella GUI Vim è in grado di riconoscere l'incollamento e
generalmente si comporterà come previsto senza bisogno di impostare
'paste'. Lo stesso si dica per un terminale nel quale Vim gestisce
direttamente i clic del mouse.
Quest'opzione è inattivata quando si lavora usando una GUI. Quindi se
la impostate nel vostro .vimrc, funzionerà in un terminale, ma non
nella GUI. Impostare 'paste' mentre si lavora usando una GUI ha degli
effetti collaterali: ad es. il bottone "Paste" nella toolbar non
funzionerà più in modo Insert, perché usa una mappatura.
Quando l'opzione 'paste' è impostata (e anche se è già attiva):
- la mappatura nei modi Insert e Command-line è inattiva
- le abbreviazioni sono inattive
- 'autoindent' è impostata a off
- 'expandtab' è impostata a off
- 'formatoptions' è usata come se fosse vuota
- 'revins' è impostata a off
- 'ruler' è impostata a off
- 'showmatch' è impostata a off
- 'smartindent' è impostata a off
- 'softtabstop' è impostato a 0
- 'textwidth' è impostato a 0
- 'wrapmargin' è impostato a 0

Le opzioni seguenti conservano il loro valore, ma non hanno
alcun effetto:
- 'cindent'
- 'indentexpr'
- 'lisp'

NOTA: Se iniziate a modificare un altro file mentre l'opzione 'paste'
è impostata, impostazioni provenienti da modeline o da autocomandi
possono cambiare ancora le impostazioni, causando problemi quando si
incolla del testo. Sarebbe buona cosa specificare di nuovo l'opzione
'paste'.
Quando l'opzione 'paste' è messa a off le opzioni sopra citate sono
ripristinate al valore che avevano subito prima che 'paste' fosse
impostato a off o a on.
Impostare a off 'paste' senza averlo prima impostato a on non produce

alcun effetto.
Poiché la mappatura non funzione se 'paste' è impostata, si deve usare l'opzione 'pastetoggle' per attivare/disattivare l'opzione 'paste' usando qualche tasto.

'pastetoggle' *'pt'*
'pastetoggle' 'pt' stringa (default: "")
globale
{non in Vi}
Se specificata, indica il tasto che attiva/disattiva l'opzione 'paste'. Questo equivale a specificare una mappatura: >
:map {tasto} :set invpaste<CR>
< Dove {tasto} è il valore di 'pastetoggle'.
La differenza è che quest'opzione è attiva anche quando 'paste' è impostato.
'pastetoggle' funziona nei modi Insert e Normal, ma non in modo Command-line.
Le mappature sono controllate in precedenza, e quindi prevalgono su 'pastetoggle'. Comunque, quando 'paste' è attivo le mappature sono ignorate in modo Insert, quindi si può fare questo: >
:map <F10> :set paste<CR>
:map <F11> :set nopaste<CR>
:imap <F10> <C-O>:set paste<CR>
:imap <F11> <nop>
:set pastetoggle=<F11>
< In questo modo premendo <F10> si inizia il modo paste e premendo <F11> si termina il modo paste.
Nota Battendo <F10> in modo Paste si inseriscono i caratteri "<F10>", poiché in modo Paste ogni cosa è inserita letteralmente, tranne la sequenza che corrisponde all'opzione 'pastetoggle'.
Quando il valore è formato da parecchi byte, viene applicata 'ttimeoutlen'.

'pex' *'patchexpr'*
'patchexpr' 'pex' stringa (default: "")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |+diff|}
Espressione da valorizzare per applicare una patch a un file e generare la risultante nuova versione del file. Vedere |diff-patchexpr|.

'patchmode' *'pm'* *E205* *E206*
'patchmode' 'pm' stringa (default: "")
globale
{non in Vi}
Se specificata, la versione più vecchia di un file è conservata. Questo si può usare per mantenere la versione originale di un file se state cambiando dei file in una distribuzione sorgente. Solo la prima volta che un file è scritto verrà conservata una copia del file originale. Il nome della copia è il nome del file originale con aggiunta la stringa specificata nell'opzione 'patchmode'. Questa opzione dovrebbe iniziare con un ".". Usare una stringa tipo ".orig" o ".org". 'backupdir' deve essere specificata perché questo funzioni. (In dettaglio: Il file di backup è rinominato col nome specificato da 'patchmode' dopo che il nuovo file è stato scritto con successo, ecco perché è necessario poter scrivere un file di backup). Se non c'è alcun file di cui fare il backup, viene creato un file vuoto. Quando il nome del file corrisponde a uno di quelli specificati con 'backupskip', un file 'patchmode' non viene creato.
Se si usa 'patchmode' per file compressi il nome di 'patchmode" viene aggiunto alla fine del nome del file originale (ad es., "file.gz.orig"), e quindi il nome che ne risulta non è sempre riconosciuto come nome di un file compresso.
Solo caratteri normali possono essere usati nei nomi di file, i caratteri "/*?[|<>" non sono validi.

'path' *'pa'* *E343* *E345* *E347* *E854*
'path' 'pa' stringa (default in Unix: ".,/usr/include,,"
in OS/2: ".,/emx/include,,"
altri sistemi: ".,,")
globale o locale al buffer |global-local|

{non in Vi}
Questa è una lista di directory che saranno scandite quando si usano i comandi |**gf**|, [f,]f, ^Wf, |:**find**|, |:**sfind**|, |:**tabfind**| e altri comandi, nel caso in cui il file che si sta cercando sia designato in maniera relativa (non iniziante con "/", "./" o "../").
Le directory specificate nell'opzione 'path' possono essere relative o assolute.
- Usare virgole per separare nomi di directory: >
:set path=.,/usr/local/include,/usr/include
< - Spazi possono anche essere usati come separatori fra nomi di directory (per compatibilità all'indietro con la versione 3.0). Per specificare uno spazio in un nome di directory, precedetelo con un backslash extra, e proteggete lo spazio [con un altro backslash]: >
:set path=.,/dir/with\\\ space
< - Per includere una virgola in un nome di directory precedetelo con un backslash extra: >
:set path=.,/dir/with\\,comma
< - Per ricercare relativamente alla directory del file corrente, usare: >
:set path=.
< - Per ricercare nella directory corrente usare una stringa nulla fra due virgole: >
:set path=,,
< - Un nome di directory può finire con ':' o '/'.
- Le variabili d'ambiente vengono valutate |:**set_env**|.
- Se si usa |**netrw.vim**| è possibile specificare degli URL. Ad es., aggiungendo "http://www.vim.org" permetterà che il comando >
:find index.html"
< funzioni.
- Ricerca verso l'alto o verso il basso di un albero di directory usando "*", "**", e ";". Vedere |**file-searching**| per informazioni e per la sintassi.
{non disponibile se compilato senza la funzionalità |**+path_extra**|}
- Attenti con i caratteri '\', battetene due per inserirne uno nell'opzione: >
:set path=.,c:\\include
< Oppure usate semplicemente '/' al posto di '\': >
:set path=.,c:/include
< Non dimenticate "." o non saranno trovati dei file neppure se sono nella stessa directory del file corrente!
La lunghezza massima è limitata. Il limite dipende dal sistema, per lo più è qualcosa tipo 256 o 1024 caratteri.
Potete controllare se tutti i file include (nel vostro programma) sono disponibili, usando il valore corrente di 'path', vedere |:**checkpath**|.
L'uso di |:**set+=**| e |:**set-=**| è da preferire quando si aggiungono o rimuovono directory dalla lista. Ciò permette di evitare problemi laddove una futura versione usasse un altro default. Per rimuovere la directory corrente usare: >
:set path-=
< Per Aggiungere la directory corrente usare: >
:set path+=
< Per usare una variabile d'ambiente, dovete probabilmente rimpiazzare il separatore. Ecco un esempio per aggiungere $INCL, una variabile in cui i nome di directory sono separati dal ";": >
:let &path = &path . "," . substitute($INCL, ';', ',', 'g')
< Sostituite il ';' con un ':' o qualsiasi separatore si usi.
Nota Questo non funziona quando $INCL contiene una virgola o uno spazio bianco.

'perldll'
'perldll' stringa (default: dipende dalla compilazione di Vim)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+perl/dyn**|}
Specifica il nome della libreria condivisa Perl. Il valore di default è DYNAMIC_PERL_DLL, che era stato specificato al momento della compilazione.
Le variabili d'ambiente sono valutate |:**set_env**|.
Quest'opzione non può essere impostata da una |**modeline**| o nel |**sandbox**|, per motivi di sicurezza.

'preserveindent' *'pi'* *'nopreserveindent'* *'nopi'*
'preserveindent' 'pi' booleana (default: off)
locale al buffer
{non in Vi}
Se viene cambiata l'indentatura della riga corrente, cercare di conservare per quanto possibile la struttura di indentatura. Normalmente l'indentatura è sostituita da una serie di <TAB>, seguiti dal numero di spazi richiesti (a meno che sia abilitata 'expandtab', nel qual caso si usano solo spazi). Abilitare quest'opzione significa che l'indentatura conserverà quanti più caratteri esistenti possibile nell'effettuare l'indentatura, e aggiungerà <TAB> o spazi in più, quanti ne sono necessari.
'expandtab' non vale per lo spazio bianco già presente, una <Tab> resta sempre una <Tab>
NOTA: Quando si usa ">>" più volte, l'indentatura risultante è un misto di <TAB> e spazi. Potreste non gradirlo.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.
Vedere anche 'copyindent'.
Usare |**:retab**| per eliminare gli spazi non necessari [verranno sostituiti da <TAB> per quanto possibile].

'previewheight' *'pvh'*
'previewheight' 'pvh' numero (default: 12)
globale
{non in Vi}
{non disponibile se compilato senza le funzionalità |**+windows**| o |**+quickfix**|}
Altezza di default per una finestra di anteprima. Usato per |**:ptag**| e comandi associati. Usato per |**CTRL-W_}**| quando non si specifica un contatore.

'previewwindow' *'nopreviewwindow'*
'pvw' *'nopvw'* *E590*
'previewwindow' 'pvw' booleana (default: off)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza le funzionalità |**+windows**| |**+quickfix**|}
Identifica la finestra di anteprima. Solo una delle finestre può avere quest'opzione impostata a on. Non viene di solito impostata direttamente, ma attraverso l'uso di uno dei comandi |**:ptag**|, |**:pedit**|, etc.

'printdevice' *'pdev'*
'printdevice' 'pdev' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+printer**|}
Il nome della stampante da usare per |**:hardcopy**|.
Vedere |**pdev-option**|.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'printencoding' *'penc'* *E620*
'printencoding' 'penc' stringa (default: "", tranne che per alcuni sistemi)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+printer**| e |**+postscript**|}
Imposta la codifica di carattere da usare nella stampa.
Vedere |**penc-option**|.

'printexpr' *'pexpr'*
'printexpr' 'pexpr' stringa (default: vedere sotto)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+printer**| e |**+postscript**|}
Espressione da valutare per stampare il file PostScript prodotto da ":hardcopy".
Vedere |**pexpr-option**|.

Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'printfont' *'pfn'* *E613*
'printfont' 'pfn' stringa (default: "courier")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+printer**|}
Quest'opzione specifica il nome del font da usare per |**:hardcopy**|. Vedere |**pfn-option**|.

'printheader' *'pheader'*
'printheader' 'pheader' stringa (default: "%<%f%h%m%=Page %N")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+printer**|}
Quest'opzione definisce il formato della testata prodotta nell'output del comando |**:hardcopy**|.
Vedere |**pheader-option**|.

'printmbcharset' *'pmbcs'*
'printmbcharset' 'pmbcs' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+printer**|, |**+postscript**| e |**+multi_byte**|}
Il font CJK da usare per l'output CJK da |**:hardcopy**|.
Vedere |**pmbcs-option**|.

'printmbfont' *'pmbfn'*
'printmbfont' 'pmbfn' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+printer**|, |**+postscript**| e |**+multi_byte**|}
Lista di nomi di font da usare per l'output CJK da |**:hardcopy**|.
Vedere |**pmbfn-option**|.

'printoptions' *'popt'*
'printoptions' 'popt' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+printer**|}
Quest'opzione è una lista di elementi che controllano il formato di output di |**:hardcopy**|:
Vedere |**popt-option**|.

'prompt' *'noprompt'*
'prompt' booleana (default: on)
globale
Quando è attiva, un prompt ":" è usato mentre si lavora in modo Ex.

'pumheight' *'ph'*
'pumheight' 'ph' numero (default: 0)
globale
{non disponibile se compilato senza la funzionalità |**+insert_expand**|}
{non in Vi}
Determina il numero massimo di elementi da visualizzare in un menù dinamico (popup) per il completamento in modo Insert. Se vale zero, si usa tutto lo spazio disponibile. |**ins-completion-menu**|.

'pumwidth' *'pw'*
'pumwidth' 'pw' numero (default: 15)
globale
{non disponibile se compilato senza la funzionalità |**+insert_expand**|}
{non in Vi}
Determina la larghezza minima da usare nel menu dinamico (popup) per il completamento in modo Insert.
|**ins-completion-menu**|.

'pythondll'
'pythondll' stringa (default: dipende dalla compilazione di Vim)
globale
{non in Vi} {solo per Unix}
{disponibile solo se compilato con la funzionalità
|+python3/dyn|}
Specifica il nome della libreria condivisa Python 2.x. Il default è
DYNAMIC_PYTHON_DLL, specificato al momento della compilazione.
Le variabili d'ambiente sono valutate |:set_env|.
Quest'opzione non può essere impostata da |modeline|, o nel |sandbox|,
per motivi di sicurezza.

'pythonhome'
'pythonhome' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+python/dyn|}
Specifica il nome della home directory di Python 2.x. Quando
l'opzione 'pythonhome' e la variabile d'ambiente PYTHONHOME non sono
impostate, verrà usato il valore PYTHON_HOME, che era stato
specificato al momento della compilazione di Vim, come nome della
home directory di Python 2.x.
Le variabili d'ambiente vengono valutate |:set_env|.
Quest'opzione non può essere impostata da una |modeline| o nel
|sandbox|, per motivi di sicurezza.

'pythonthreedll'
'pythonthreedll' stringa (default: dipende dalla compilazione di Vim)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+python3/dyn|}
Specifica il nome della libreria condivisa Python 3. Il default è
DYNAMIC_PYTHON3_DLL, specificato al momento della compilazione.
Le variabili d'ambiente sono valutate |:set_env|.
Quest'opzione non può essere impostata da |modeline|, o nel |sandbox|,
per motivi di sicurezza.

'pythonthreehome'
'pythonthreehome' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+python3/dyn|}
Specifica il nome della home directory di Python 3. Quando l'opzione
'pythonthreehome' e la variabile d'ambiente PYTHONHOME non sono
impostate, verrà usato il valore PYTHON3_HOME, che era stato
specificato al momento della compilazione di Vim, come nome della
home directory di Python 3.
Le variabili d'ambiente vengono valutate |:set_env|.
Quest'opzione non può essere impostata da una |modeline| o nel
|sandbox|, per motivi di sicurezza.

'pyxversion' *'pyx'*
'pyxversion' 'pyx' numero (il default varia secondo le compilazioni)
globale
{non in Vi}
{disponibile solo se compilato con le funzionalità
|+python| o |+python3}
Specifica la versione di python usata per le funzioni e i comandi
|python_x|. Il valore di default è il seguente:

Vim compilato con	Default ~
|+python| e |+python3|	0
solo |+python|	2
solo |+python3|	3

I valori che si possono specificare sono solo 0, 2 e 3.
Se 'pyxversion' vale 0, è impostata a 2 o 3 dopo che è la prima
esecuzione di un comando o di una funzione python2/3. P.es.
immettendo `:py` il valore viene impostato a 2, mentre con `:py3`

viene impostato a 3. L'esecuzione di`:pyx`lo imposta a 3 se
Python 3 è disponibile, altrimenti lo imposta a 2 se è disponiibile
Python 2.
Vedere anche: |**has-pythonx**|

Se Vim è compilato solo con la funzionalità |**+python**| o |**+python3**|
impostare 'pyxversion' non produce effetto alcuno. Le funzioni e i
comandi pyx* coincidono sempre con quelli contenuti nella versione
compilata.

Quest'opzione non può essere impostata da una |**modeline**| o nel
|**sandbox**|, per motivi di sicurezza.

'quoteescape' *'qe'*
'quoteescape' 'qe' stringa (default: "\")
locale al buffer
{non in Vi}
I caratteri da usare per proteggere i vari tipi di virgolette in una
stringa. Usata per oggetti come a', a" e a` |**a'**|.
Quando uno dei caratteri specificati in quest'opzione è presente
all'interno di una stringa, il carattere seguente verrà ignorato.
Il valore di default fa sì che un testo come "foo\"bar\\" sia
considerato come una stringa.

'readonly' *'ro'* *'noreadonly'* *'noro'*
'readonly' 'ro' booleana (default: off)
locale al buffer
Quando è impostata a on, la scrittura non viene eseguita, a meno che
non si specifichi un '!'. Serve come protezione per evitare di
sovrascrivere per sbaglio un file. Attiva per default quando Vim è
invocato in modalità sola lettura ("vim -R") o quando vim viene
invocato con il nome "view".
Quando si usa ":w!" l'opzione 'readonly' è annullata per il buffer
corrente, a meno che in 'cpoptions' sia specificato il flag 'Z'.
{non in Vi} Quando si sta usando il comando ":view" l'opzione
'readonly' è impostata per ogni nuovo buffer aperto.
Vedere 'modifiable' per impedire modifiche al buffer.

'redrawtime' *'rdt'*
'redrawtime' 'rdt' numero (default: 2000)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|**+reltime**|}
Il tempo necessario, in millisecondi, per ridisegnare la schermata.
Applicabile per l'evidenziazione nella ricerca di espressioni regolari
con 'hlsearch', per l'evidenziazione con |**:match**| e per
l'evidenziazione sintattica.
Quando rinfrescare la schermata richiede più di questo numero di
millisecondi, non verranno evidenziati ulteriori corrispondenze.
Per l'evidenziazione sintattica, il tempo vale per ogni singola
finestra. Quando lo si supera, l'evidenziazione sintattica è
disabilitata, finché non viene dato un comando |**CTRL-L**|.
Questo serve a evitare che Vim si blocchi durante la ricerca di
un'espressione molto complicata.
'regexpengine' *'re'*
'regexpengine' 're' numero (default: 0)
globale
{non in Vi}
Quasta opzione permettere di scegliere quale codice utilizzare per
default per effettuare la valutazione delle espressione regolari.
|**two-engines**|
I possibili valori sono:
0 selezione automatica
1 codice tradizionale [vecchio]
2 codice NFA [nuovo]
Nota Quando si usa il codice NFA e l'espressione da valutare contiene
qualcosa che non è supportato l'espressione risulterà senza alcuna
corrispondenza. Questo è utile solo per effettuare il debug del nuovo
codice di valutazione [regexp engine].
Usare la selezione automatica consente a Vim di cambiare codice, se
il codice di default consuma troppe risorse. Per esempio quando il
codice NFA genera troppi stati [troppe alternative da controllare].

Ciò dovrebbe impedire che Vim resti bloccato esaminando una
combinazione di una stringa di ricerca complessa con un testo lungo.

'relativenumber' *'rnu'* *'norelativenumber'* *'nornu'*
'relativenumber' 'rnu' booleana (default: off)
locale alla finestra
{non in Vi}
Viene mostrato davanti a ogni riga il numero di riga rispetto alla
riga su cui sulla quale sta il cursore. La numerazione facilita
l'utilizzo di |**count**| (il contatore). Potete anteporlo ad alcuni
comandi di movimento verticale (per esempio j k + -) senza dovelo
calcolare da soli. Specialmente utile se usato con altro comandi (per
esempio y d c < > gq gw =).
Quando l'opzione 'n' è esclusa da 'cpoptions', una riga lunga non
utilizzerà le colonne destinate a contenere i numeri di riga (è questo
il default quando 'compatible' non è impostato).
L'opzione 'numberwidth' può essere usata per impostare il numero di
colonne da utlizzare per la numerazione delle righe.
Quando una riga lunga, visualizzata sono in parte, non è visibile
dall'inizio, dei caratteri '-' sono posti prima del numero.
Vedere |**hl-LineNr**| e |**hl-CursorLineNr**| per l'evidenziazione usata
per i numeri.

Il numero che precede la linea su cui sta il cursore dipende anche dal
valore dell'opzione 'number', vedere |**number_relativenumber**| per tutte
le possibili combinazione delle due opzioni.

'remap' *'noremap'*
'remap' booleana (default: on)
globale
Consente la ricorsività nelle mappature. Se non volete questa
opzione solo per un singolo comando, usate il comando: :noremap[!].
NOTA: Per evitare problemi di portabilità con script Vim, mantenere
sempre quest'opzione al valore di default "on". Metterla a "off" solo
se si usano vecchi script Vi.

'renderoptions' *'rop'*
'renderoptions' 'rop' stringa (default: "")
globale
{non in Vi}
{disponibile solo se compilato con GUI e DIRECTX in
MS-Windows}
Scegliere un renderizzatore di testo e impostarne le opzioni. Le
opzioni dipendono dal renderizzatore.

Sintassi: >
set rop=type:{renderizzatore}(,{nome}:{valore})*
<
Attualmente è disponibile un solo renderizzatore opzionale.

render. comportamento ~
directx Vim visualizzerà il testo usando DirectX (DirectWrite).
I glifi visualizzati in questo modo sono più eleganti che
usando GDI, il renderizzatore di default.
L'opzione 'encoding' dev'essere impostata a "utf-8", ed
è necessario che la versione di MS-Windows sia Vista o una
versione successiva.

Opzioni:

nome	significato	tipo	valore ~
gamma	gamma	decim.	1.0 - 2.2 (forse)
contrast	enhancedContrast	decim.	(ignoto)
level	clearTypeLevel	decim.	(ignoto)
geom	pixelGeometry	int.	0 - 2 (vedi sotto)
renmode	renderingMode	int.	0 - 6 (vedi sotto)
taamode	textAntialiasMode	int.	0 - 3 (vedi sotto)
scrlines	Scroll Lines	int.	(deprecato)

Vedere questa pagina Web per ulteriori dettagli
(ma non riguardo a scrlines):
https://msdn.microsoft.com/en-us/library/dd368190.aspx

Per geom: struttura di un pixel del dispositivo.

```
                  0 - DWRITE_PIXEL_GEOMETRY_FLAT
                  1 - DWRITE_PIXEL_GEOMETRY_RGB
                  2 - DWRITE_PIXEL_GEOMETRY_BGR

                Vedere questa pagina Web per ulteriori dettagli:
                  https://msdn.microsoft.com/en-us/library/dd368114.aspx

                Per renmode: metodo di renderizzazione dei glifi.
                  0 - DWRITE_RENDERING_MODE_DEFAULT
                  1 - DWRITE_RENDERING_MODE_ALIASED
                  2 - DWRITE_RENDERING_MODE_GDI_CLASSIC
                  3 - DWRITE_RENDERING_MODE_GDI_NATURAL
                  4 - DWRITE_RENDERING_MODE_NATURAL
                  5 - DWRITE_RENDERING_MODE_NATURAL_SYMMETRIC
                  6 - DWRITE_RENDERING_MODE_OUTLINE

                Vedere questa pagina Web per ulteriori dettagli:
                  https://msdn.microsoft.com/en-us/library/dd368118.aspx

                Per taamode: tipo di antialiasing usato per disegnare testo.
                  0 - D2D1_TEXT_ANTIALIAS_MODE_DEFAULT
                  1 - D2D1_TEXT_ANTIALIAS_MODE_CLEARTYPE
                  2 - D2D1_TEXT_ANTIALIAS_MODE_GRAYSCALE
                  3 - D2D1_TEXT_ANTIALIAS_MODE_ALIASED

                Vedere questa pagina Web per ulteriori dettagli:
                  https://msdn.microsoft.com/en-us/library/dd368170.aspx

                Per scrlines:
                Questo valore era usato per ottimizzare il comportamento
                dello scorrimento, tuttavia è divenuto deprecato.  Se lo si
                specifica, viene semplicemente ignorato.

                Esempio: >
                  set encoding=utf-8
                  set gfn=Ricty_Diminished:h12
                  set rop=type:directx
<
                Impostando un carattere "raster" [a reticolo] (Courier,
                Terminal o FixedSys, che hanno come suffisso ".fon" nel
                nome del file) come 'guifont', esso sarà disegnato usando
                la GDI [Graphic Device Interface di Windows] come metodo
                di ripiego.

                NOTA: È cosa nota che la combinazione di alcuni caratteri
                con alcune opzioni può provocare difficioltà nel disegno
                di glifi.

                  - 'renmode:5' e 'renmode:6' non funzionano con alcuni
                    caratteri di fattura speciale (caratteri True-Type che
                    includono solo glifi descritti con una mappa di bit
                    [bitmap]).
                  - 'taamode:3' non funziona con alcuni caratteri di tipo
                    vettoriale.

                NOTA: Per mezzo di quest'opzione, è possibile visualizzare
                emoji (emoticon, simboli pittografici) colorati, in Windows
                8.1 e versioni successive.  Per visualizzare emoji colorati,
                devono essere soddisfatte alcune condizioni, di cui occorre
                prendere nota.

                  - Se il carattere [font] in uso include già emoji non
                    colorati, continuerà ad essere utilizzato.
                  - Se il carattere in uso non prevede emoji, il sistema
                    sceglie un carattere alternativo capace di rappresentare
                    emoji.  In ambiente Windows 10, sarà usato il carattere
                    "Segoe UI Emoji".
                  - Quando un tale carattere alternativo non sia dotato di
                    glifi a larghezza fissa, gli emoji potrebbero essere
                    rappresentati superando il limite di spazio assegnato al
                    rettangolo che delimita lo spazio in cui viene disegnato
                    un carattere.
```

Altri tipi di renderizzatori non sono attualmente supportati.

'report'
'report' numero (default: 2)
globale
Soglia per ottenere il messaggio sul numero di righe modificate.
Quando il numero di righe modificate è superiore a 'report' un
messaggio sarà dato per buona parte dei comandi ":". Per vedere
sempre il messaggio, impostare 'report' a 0.
Per il comando ":substitute" il numero di sostituzioni effettuate è
usato al posto del numero di righe modificate.

'restorescreen' *'rs'* *'norestorescreen'* *'nors'*
'restorescreen' 'rs' booleana (default: on)
globale
{non in Vi}
{solo nella versione console di Windows 95/NT}
Quando impostato a on, il contenuto dello schermo è ripristinato
all'uscita da Vim. Ciò viene fatto anche quando si eseguono comandi
esterni.

Per Vim non-Windows Vim: Si possono impostare o modificare le opzioni
't_ti' e 't_te' nel vostro .vimrc. Per inibire il ripristino:
set t_ti= t_te=
Per abilitare il ripristino (per un xterm):
set t_ti=^[7^[[r^[[?47h t_te=^[[?47l^[8
(Dove ^[è il tasto <Esc>, battere CTRL-V <Esc> per inserirlo)

'revins' *'ri'* *'norevins'* *'nori'*
'revins' 'ri' booleana (default: off)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|**+rightleft**|}
L'immissione caratteri in modo Insert funziona a rovescio. Vedere
come immettere all'indietro ["typing backwards"] |**ins-reverse**|.
Quest'opzione può essere attivata/inattivata con il comando CTRL-_
in modo Insert, se 'allowrevins' è impostato a on.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible' o
'paste'.
Quest'opzione è impostata a off quando si imposta 'paste' e
ripristinata quando 'paste' è messa a off.

'rightleft' *'rl'* *'norightleft'* *'norl'*
'rightleft' 'rl' booleana (default: off)
locale alla finestra
{non in Vi}
{disponibile solo se compilato con la funzionalità
|**+rightleft**|}
Se impostata, l'orientamento del video diventa da destra a sinistra,
ossia i caratteri che sono contenuti nel file sono visualizzati da
destra a sinistra.
Usando quest'opzione, è possibile editare file per lingue che
sono scritte da destra a sinistra, come l'Ebraico e l'Arabo.
Quest'opzione è attivata a livello di finestra singola, e quindi è
possibile editare file misti nella stessa sessione, o vedere lo stesso
file in entrambe le direzioni (questo può tornare utile quando avete
un testo misto, che contiene righe in entrambe le direzioni, che
possono così essere visualizzate correttamente in finestre differenti.
Vedere anche |**rileft.txt**|.

'rightleftcmd' *'rlc'*
'rightleftcmd' 'rlc' stringa (default: "search")
locale alla finestra
{non in Vi}
{disponibile solo se compilato con la funzionalità
|**+rightleft**|}
Ogni parola in quest'opzione permette di modificare in modalità "da
destra a sinistra" un gruppo di comandi:

search comandi "/" e "?"

Ciò è utile per lingue come l'Ebraico, l'Arabo e il Farsi.

L'opzione 'rightleft' deve essere attiva per poter utilizzare
'rightleftcmd'.

'rubydll'
'rubydll' stringa (default: dipende dalla compilazione di Vim)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|**+ruby/dyn**|}
Specifica il nome della libreria condivisa Ruby. Il default è
DYNAMIC_RUBY_DLL, specificato al momento della compilazione.
Le variabili d'ambiente sono valutate |**:set_env**|.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|,
per motivi di sicurezza.

'ruler' *'ru'* *'noruler'* *'noru'*
'ruler' 'ru' booleana (default: off, impostata a on in
|**defaults.vim**|)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+cmdline_info**|}
Mostra il numero di riga e di colonna in cui si trova il cursore,
separate da una virgola. Se c'è posto, la posizione relativa del testo
visualizzato è visibile all'estrema destra:
Cim la prima riga è visibile
Fon l'ultima riga è visibile
Tut sia la prima che l'ultima riga sono visibili
45% posizione relativa nel file
Se 'rulerformat' è impostato, detta il formato del ruler [letteralmente
righello]. Ogni finestra ha il suo righello. Se la finestra ha una
riga di status, il righello è mostrato al suo interno. Altrimenti è
visibile sull'ultima riga dello schermo. Se la riga di status è
specificata tramite 'statusline' (ossia non è la stringa nulla),
quest'opzione prevale su 'ruler' e 'rulerformat'.
Se il numero di caratteri visualizzato sullo schermo è diverso dal
numero di byte nel testo corrispondente (ad es., per un carattere
<TAB> o per un carattere multi-byte), sia la colonna nel testo
(numero di byte) che la colonna sullo schermo sono visualizzate,
separate da un trattino.
Una riga vuota è descritta come "0-1".
Per un buffer vuoto anche il numero di riga sarà a zero: "0,0-1".
Quest'opzione è messa a off quando si imposta l'opzione 'paste'
e ripristinata quando 'paste' è messa a off.
Se non volete vedere il righello durante tutto l'edit, ma volete
sapere a che punto siete del file, usate "g CTRL-G" |**g_CTRL-G**|.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'rulerformat' *'ruf'*
'rulerformat' 'ruf' stringa (default: "")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+status_line**|}
Quando quest'opzione non è nulla, determina il contenuto della stringa
del righello, come visualizzata dall'opzione 'ruler'.
Il formato di quest'opzione è come quello di 'statusline'.
La larghezza di default del righello è di 17 caratteri. Per avere un
righello di 15 caratteri, mettere "%15(" all'inizio e "%)" alla fine.
Esempio: >
:set rulerformat=%15(%c%V\ %p%%)
<
'runtimepath' *'rtp'* *vimfiles*
'runtimepath' 'rtp' stringa (default:
Unix: "$HOME/.vim,
$VIM/vimfiles,
$VIMRUNTIME,
$VIM/vimfiles/after,
$HOME/.vim/after"
Amiga: "home:vimfiles,
$VIM/vimfiles,
$VIMRUNTIME,
$VIM/vimfiles/after,

```
                                        home:vimfiles/after"
                                PC, OS/2: "$HOME/vimfiles,
                                        $VIM/vimfiles,
                                        $VIMRUNTIME,
                                        $VIM/vimfiles/after,
                                        $HOME/vimfiles/after"
                                Macintosh: "$VIM:vimfiles,
                                        $VIMRUNTIME,
                                        $VIM:vimfiles:after"
                                RISC-OS: "Choices:vimfiles,
                                        $VIMRUNTIME,
                                        Choices:vimfiles/after"
                                VMS: "sys$login:vimfiles,
                                        $VIM/vimfiles,
                                        $VIMRUNTIME,
                                        $VIM/vimfiles/after,
                                        sys$login:vimfiles/after")
                        globale
                        {non in Vi}
        Questa è una lista di directory che sarà utilizzata per trovare
        dei file utilizzati al momento dell'esecuzione di Vim:
          filetype.vim  tipo file dal nome file |new-filetype|
          scripts.vim   tipo file dal contenuto file |new-filetype-scripts|
          autoload/     script caricati automaticamente |autoload-functions|
          colors/       file con schemi di colorazione |:colorscheme|
          compiler/     file del compilatore |:compiler|
          doc/          documentazione |write-local-help|
          ftplugin/     plugin per tipo file |write-filetype-plugin|
          indent/       script di indentatura |indent-expression|
          keymap/       file mappatura tastiera |mbyte-keymap|
          lang/         traduzione del menù |:menutrans|
          menu.vim      menù per le varie GUI |menu.vim|
          pack/         pacchetti di plugin |:packadd|
          plugin/       script di plugin |write-plugin|
          print/        file per stampare |postscript-print-encoding|
          spell/        file di correzione ortografica |spell|
          syntax/       file per la colorazione sintattica |mysyntaxfile|
          tutor/        file per comando vimtutor |tutor|

        E ogni altro file che può essere cercato per mezzo del comando
        |:runtime|.

        I default per la maggior parte dei sistemi cercano in cinque posti:
        1. Nella vostra home directory, per le vostre preferenze personali.
        2. In una directory comune a tutto il sistema, per preferenze
           impostate dall'amministratore di sistema.
        3. In $VIMRUNTIME, per file distribuiti con Vim.
                                                        *after-directory*
        4. Nella directory "after" ("ulteriore") contenuta nella directory
           Vim comune a tutto il sistema.  Questa serve all'amministratore
           del sistema per forzare scelte, o per aggiungere qualcosa ai
           default impostati nella distribuzione (è usata raramente).
        5. Nella directory "after" ("ulteriore") contenuta nella vostra home
           directory.  Questa serve per forzare scelte o aggiungere qualcosa
           ai default impostati nella distribuzione o alle impostazioni valide
           a livello di sistema (raramente necessaria).

        Ulteriori elementi sono aggiunti se si usano pacchetti |packages|.
        Se ciò richiede troppo tempo, il comando `:set rtp` sarà interrotto,
        si usi `:echo &rtp` per visualizzare la stringa completa.

        Nota A differenza che per l'opzione 'path', non si possono specificare
        espressioni regolari del tipo "**".  Le espressioni regolari normali
        sono consentite, ma possono rallentare notevolmente la ricerca dei
        file di tipo "runtime".  Per velocizzare, usare il numero minimo di
        elementi necessario, ed evitare per quanto possibile l'uso di
        espressioni regolari.
        Vedere |:runtime|.
        Esempio: >
                :set runtimepath=~/vimruntime,/mygroup/vim,$VIMRUNTIME
<       In questo esempio si richiede di utilizzare per prima la directory
        "~/vimruntime" (che contiene i vostri file personali di Vim da usare
        al "runtime"), poi "/mygroup/vim" (condivisa da un gruppo di persone
```

e infine "$VIMRUNTIME" (i file di tipo "runtime" contenuti nella
distribuzione di Vim).
Sarebbe probabilmente meglio includere $VIMRUNTIME da qualche parte,
per usare i file di tipo "runtime" contenuti nella distribuzione di
Vim. Si può mettere una directory prima di $VIMRUNTIME per trovare
file che sostituiscano un file di tipo "runtime" contenuto nella
distribuzione. Si può mettere una directory dopo $VIMRUNTIME per
trovare file da aggiungere a quelli che si trovano nella distribuzione
di Vim.
Quando Vim è chiamato con l'opzione |**--clean**| gli elementi contenuti
nell'home directory non sono inclusi.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|,
per motivi di sicurezza.

'scroll' *'scr'*
'scroll' 'scr' numero (default: metà dell'altezza della finestra)
locale alla finestra
Numero di righe di scorrimento da usare coi comandi CTRL-U e CTRL-D.
Sarà impostata alla metà del numero di righe nella finestra quando la
dimensione della finestra viene cambiata. Se specificate un contatore
ai comandi CTRL-U o CTRL-D sarà usato come nuovo valore di 'scroll'.
Potete re-impostarlo al valore di metà altezza della finestra col
comando ":set scroll=0". {Vi è leggermente diverso: 'scroll'
specifica il numero di righe sullo schermo, invece che il numero di
righe nel file, la differenza si nota quando una riga del file è
visualizzata su più di una riga dello schermo}

'scrollbind' *'scb'* *'noscrollbind'* *'noscb'*
'scrollbind' 'scb' booleana (default: off)
locale alla finestra
{non in Vi}
Vedere anche |**scroll-binding**|. Quando quest'opzione è impostata, la
finestra corrente scorre come le altre finestre "scroll-bind" (quelle
per le quali è stata impostata quest'opzione). Quest'opzione è utile
per visualizzare le differenze fra due versioni di un file, vedere
'diff'.
Vedere |'**scrollopt**'| per opzioni che determinano il modo in cui questa
opzione va interpretata.
Quest'opzione è quasi sempre messa a off quando si divide una
finestra per editare un altro file. Questo vuol dire che il comando:
":split | edit file" genera due finestre entrambe di tipo
"scroll-bind", ma ":split file" non genera una finestra di tipo
"scroll-bind".

'scrolljump' *'sj'*
'scrolljump' 'sj' numero (default: 1)
globale
{non in Vi}
Numero minimo di righe di scorrimento da usare quando il cursore
"esce" dallo schermo (ad es. col comando "j"). Non usata coi comandi
di scorrimento (ad es., CTRL-E, CTRL-D). Utile se il vostro terminale
scorre molto lentamente.
Quando è impostato a un numero negativo da -1 a -100, questo viene
considerato come una percentuale dell'altezza della finestra. Quindi
-50 fa scorrere la finestra di una metà della sua altezza.
NOTA: Quest'opzione è impostata a on se si attiva 'compatible'.

'scrolloff' *'so'*
'scrolloff' 'so' numero (default: 0, impostata a 5 in |**defaults.vim**|)
globale
{non in Vi}
Numero minimo di righe dello schermo da mostrare sopra e sotto il
cursore. Serve a rendere visibile un po' di contesto attorno al punto
in cui si sta lavorando. Se impostato a un valore molto alto (999),
la riga che contiene il cursore sarà sempre nel mezzo della finestra
(tranne che quando si lavora a inizio o fine file, o se righe lunghe
del file occupano più righe dello schermo.
Per lo scorrimento orizzontale vedere 'sidescrolloff'.
NOTA: Quest'opzione è impostata a 0 se si attiva 'compatible'.

'scrollopt' *'sbo'*
'scrollopt' 'sbo' stringa (default: "ver,jump")
globale

{non in Vi}
Una lista di parole, separate da virgola, che dice come dovrebbero comportarsi le finestre 'scrollbind'. 'sbo' significa "ScrollBind Options".
Le seguenti parole sono specificabili:
ver Sincronizza lo scorrimento verticale per le finestre 'scrollbind'.
hor Sincronizza lo scorrimento orizzontale per le finestre 'scrollbind'.
jump Serve per aggiustare lo scorrimento verticale fra due finestre. Questo numero è la differenza rispetto alla prima riga visualizzata nella finestra sincronizzata. Mentre ci si sposta all'interno di una finestra, un'altra finestra di tipo 'scrollbind' può raggiungere una posizione prima dell'inizio o dopo la fine del buffer. Lo spostamento non viene modificato, anche se, tornando indietro, la finestra di tipo 'scrollbind' verrà fatta scorrere, per quanto possibile, alla posizione desiderata.
Se quest'ultima finestra diventa poi la finestra corrente, due cose possono succedere con lo scorrimento relativo:
1. Se "jump" (salto) non è specificato, lo scorrimento relativo è modificato secondo la posizione di scorrimento nella finestra divenuta corrente. Quando si ritorna nella finestra precedente, si userà il nuovo spostamento relativo.
2. Quando "jump" è specificato, le altre finestre sono fatte scorrere per mantenere lo stesso spostamento relativo. Quando si ritorna nella finestra precedente, si usa ancora lo stesso spostamento relativo.

Vedere anche |**scroll-binding**|.
Quando si lavora in modo Diff, c'è sempre uno scorrimento verticale, anche se non si è specificato "ver".

'sections' *'sect'*
'sections' 'sect' stringa (default: "SHNHH HUnhsh")
globale
Specifica le macro del comando Unix nroff che dividono sezioni di un testo. Si tratta di coppie di lettere (vedere |**object-motions**|). Il default fa prendere come inizio di sezione le macro di "nroff" ".SH", ".NH", ".H", ".HU", ".nh" e ".sh".

'secure' *'nosecure'* *E523*
'secure' booleana (default: off)
globale
{non in Vi}
Se impostata, ":autocmd", comandi di scrittura e l'uso della shell non sono consentiti nei file ".vimrc" e ".exrc" della directory corrente e i comandi di mappatura sono visualizzati. Va impostata a off solo se si è certi di non incorrere in problemi, o quando l'opzione 'exrc' è impostata a off. In ambiente Unix quest'opzione è usata soltanto se i file ".vimrc" o ".exrc" non vi appartengono. Ciò può creare problemi solo se il sistema consente agli utenti di eseguire un comando "chown" [che cambia l'appartenenza di un file]. In questo caso è meglio impostare a on 'secure' alla fine del vostro file ~/.vimrc.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'selection' *'sel'*
'selection' 'sel' stringa (default: "inclusive")
globale
{non in Vi}
Quest'opzione specifica il comportamento alla selezione. È usato solo nei modi Visual e Select.
Valori possibili:

valore	dopo riga	inclusive ~
old	no	yes
inclusive	yes	yes
exclusive	yes	no

"dopo riga" significa che il cursore può essere posizionato nel carattere che viene dopo la fine della riga.
"inclusive" significa che l'ultimo carattere della selezione è compreso nella selezione effettuata. Ad esempio, quando "x" è usato per cancellare il testo selezionato.
Quando si specifica "old" e 'virtualedit' permette al cursore di posizionarsi oltre la fine della riga, il carattere di fine riga non è comunque incluso nella selezione.
Nota Quando si usa "exclusive" e si sceglie dal fondo andando indietro, non si può includere l'ultimo carattere di una riga, quando si inizia in modo Normale e 'virtualedit' è impostato alla stringa nulla.

L'opzione 'selection' è impostata dal comando |**:behave**|.

'selectmode' *'slm'*
'selectmode' 'slm' stringa (default: "")
globale
{non in Vi}
Una lista di parole separata da virgola, che specifica quando passare in modo Select invece che in modo Visual, all'inizio di una selezione.
Valori possibili:

mouse	quando si una il mouse
key	quando si usano tasti maiuscoli speciali
cmd	quando si usa "v", "V" o CTRL-V

Vedere |**Select-mode**|.
L'opzione 'selectmode' è impostata dal comando |**:behave**|.

'sessionoptions' *'ssop'*
'sessionoptions' 'ssop' stringa (default: "blank,buffers,curdir,folds,
! help,options,tabpages,winsize,terminal")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+mksession**|}
Modifica il comportamento del comando |**:mksession**|. È un elenco di parole separate da virgola. Ogni parola è una richiesta di salvataggio e (successivo) ripristino di qualcosa:

parola	salva e ripristina ~
blank	una finestra vuota
buffers	buffer nascosti e scaricati, non solo quelli visibili nella finestra
curdir	la directory corrente
piegature	piegature di tipo manuale, piegature aperte/chiuse e opzioni di piegatura
globals	variabili globali il cui nome inizi con lettera maiuscola e che contengano almeno una lettera minuscola. Sono memorizzate solo quelle di tipo "stringa" e "numero".
help	la finestra di aiuto
localoptions	opzioni e mappature applicabili solo a una finestra o a un buffer (non i valori globali delle opzioni locali)
options	tutte le opzioni e mappature (compresi i valori globali delle opzioni locali)
resize	dimensione della finestra Vim: 'lines' e 'columns'
sesdir	la directory in cui si trova il file di questa sessione diventerà la directory corrente (utile in caso di progetti situati in una rete formata da diversi sistemi)
slash	i backslash ("\") nei nomi di file sono da sostituire con normali barre ("/")
tabpages	tutte le linguette; senza questo parametro, solo la linguetta corrente è ripristinata, per permettervi di preparare una sessione distinta per ogni linguetta presa individualmente
terminal	incluse le finestre di terminale in cui il comando può essere ripristinato
unix	il formato di fine-riga Unix (un solo carattere <NL>), va usato anche in Windows o DOS
winpos	posizione dell'intera finestra Vim
winsize	dimensioni delle finestre

Non specificare "curdir" e "sesdir" assieme.
Quando non si specifica né "curdir" né "sesdir", i nomi di file sono memorizzati in formato assoluto (nomi "lunghi").
"slash" e "unix" sono utili on Windows quando si condividono i file in modifica con dei sistemi Unix. La versione Unix di Vim non può eseguire degli script che siano in formato DOS, ma la versione Windows di Vim può eseguire degli script in formato Unix.

'shell' *'sh'* *E91*
'shell' 'sh' stringa (default: $SHELL o "sh",
MS-DOS e Win32: "command.com" o
"cmd.exe", OS/2: "cmd")
globale
Nome della shell da usare per i comandi ! e :! . Se cambiate questo valore, controllate anche le seguenti opzioni: 'shelltype', 'shellpipe', 'shellslash' 'shellredir', 'shellquote', 'shellxquote' e 'shellcmdflag'.
È possibile fornire un argomento al comando, ad es. "csh -f".
Vedere |**option-backslash**| per inserire spazi e backslash.
Le variabili d'ambiente vengono valutate |**:set_env**|.

Se il nome della shell contiene uno spazio, potrebbe essere necessario inserirlo fra doppi apici, o proteggere lo spazio. Ad es.: >
:set shell=\"c:\program\ files\unix\sh.exe\"\ -f
< Notare il backslash davanti a ogni doppio apice, (per evitare di farlo considerare come un commento). Notare anche che "-f" non è fra doppi apici, perché non fa parte del nome del comando. Vim riconosce auto-magicamente i backslash che fanno da separatore nel nome del file. Esempio con la protezione di uno spazio (Vim si comporta così quando inizializza un'eventuale opzione contenuta in $SHELL): >
:set shell=/bin/con\\\ spazio/sh
< Il valore attribuito a 'shell' è "/bin/con\ spazio/sh", due backslash sono "consumati" da `:set`.

In MS-Windows, quando il nome del file eseguibile termina per ".com" il suffisso va necessariamente inserito. Quindi impostare la shell a "command.com" o "4dos.com" va bene, mentre impostarla a "command" o "4dos" non funziona per tutti i comandi (ad es., quelli di filtro). Per ragioni ignote, se si usa "4dos.com" la directory corrente è cambiata a "C:\". Per evitarlo, impostare 'shell' così: >
:set shell=command.com\ /c\ 4dos
< Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'shellcmdflag' *'shcf'*
'shellcmdflag' 'shcf' stringa (default: "-c";
MS-DOS e Win32, quando il nome
della 'shell' non contiene la stringa
"sh" da qualche parte: "/c")
globale
{non in Vi}
Flag passata alla shell che esegue comandi "!" e ":!" ; ad es., "bash.exe -c ls" o "command.com /c dir". Per i sistemi del tipo di MS-DOS, il default è impostato a seconda del valore di 'shell', per ridurre la necessità di dover impostare quest'opzione da parte dell'utilizzatore.
In Unix può essere presente più di un flag. Ogni parte delimitata da uno spazio è passata come argomento al comande della shell.
Vedere |**option-backslash**| per inserire spazi e backslash.
Vedere |**dos-shell**| per MS-DOS ed MS-Windows.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'shellpipe' *'sp'*
'shellpipe' 'sp' stringa (default: ">", "| tee", "|& tee" o "2>&1| tee")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+quickfix**|}
Stringa da usare ridirigere l'output del comando ":make" nel file che contiene gli errori. Vedere anche |**:make_makeprg**|.
Vedere |**option-backslash**| per inserire spazi e backslash.

Il nome del file temporaneo può essere rappresentato da "%s" se necessario (il nome del file è aggiunto automaticamente in fondo se la stringa %s non compare nel valore di quest'opzione).
Per Amiga e MS-DOS il default è ">". L'output è salvato direttamente in un file e non viene contemporaneamente visualizzato sullo schermo. Per Unix il default è "| tee". Lo "stdout" del compilatore è salvato in un file e visualizzato sullo schermo. Se l'opzione 'shell' vale "csh" oppure "tcsh", dopo le inizializzazioni il default diviene "|& tee".
Se l'opzione 'shell' vale "sh", "ksh", "mksh", "pdksh", "zsh" o "bash" il default diviene "2>&1| tee". Questo vuol dire che anche "stderr" viene incluso con "stdout". Prima di usare l'opzione 'shell', il nome della directory viene rimosso, quindi "/bin/sh" invoca "sh".
L'inizializzazione di quest'opzione è fatta dopo aver letto il file ".vimrc" e le altre inizializzazioni, così che quando l'opzione 'shell' è stata impostata in precedenza, l'opzione 'shellpipe' cambia automaticamente, a meno che sia stata impostata esplicitamente in precedenza.
Quando 'shellpipe' è impostata alla stringa nulla, non verrà effettuata nessuna ridirezione dell'output di ":make". Ciò è utile se usate un 'makeprg' che scriva già a 'makeef' per conto suo.
Se non desiderate questa ridirezione, ma volete includere 'makeef', impostate 'shellpipe' a un singolo carattere di spazio.
Non dimenticate di premettere un backslash allo spazio: ":set sp=\ ".
In futuro le ridirezioni (pipes) possono essere usate per filtrare e quest'opzione diverrà obsoleta (almeno per Unix).
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'shellquote' *'shq'*

'shellquote' 'shq' stringa (default: ""; MS-DOS e Win32, quando 'shell' contiene la stringa "sh" da qualche parte: "\"")
globale
{non in Vi}

Il carattere o i caratteri fra i quali viene incluso il comando passato alla shell, per i comandi "!" e ":!" . La ridirezione viene mantenuta fuori dal carattere (o dai caratteri) scelti. Vedere 'shellxquote' per includere la ridirezione. È probabilmente inutile impostare entrambe le opzioni contemporaneamente.
Questa è una stringa nulla per default. È nota essere utile per delle shell prodotte da terze parti, su sistemi di tipo MD-DOS, come la Korn Shell della MKS, o la bash, in cui dovrebbe valere "\"". Il default è adattato a seconda del valore di 'shell', per ridurre la necessità di impostare quest'opzione da parte dell'utilizzatore.
Vedere |**dos-shell**|.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'shellredir' *'srr'*

'shellredir' 'srr' stringa (default: ">", ">&" o ">%s 2>&1")
globale
{non in Vi}

Stringa da usare ridirigere l'output di un comando di filtro in un file temporaneo. Vedere anche |**:!**|.
Vedere |**option-backslash**| per inserire spazi e backslash.
Il nome del file temporaneo può essere rappresentato da "%s" se necessario (il nome del file è aggiunto automaticamente in fondo se la stringa %s non compare nel valore di quest'opzione).
il default è ">". Per Unix, se l'opzione 'shell' vale "csh", "tcsh" o "zsh" durante le inizializzazioni, il default diviene ">&".
Se l'opzione 'shell' vale "sh", "ksh" o "bash" il default diviene ">%s 2>&1". Questo vuol dire che anche "stderr" viene incluso con "stdout".
Per Win32, vengono effettuati gli stessi controlli che per Unix e inoltre si controlla "cmd", per il quale il default diviene ">%s 2>&1". Inoltre, gli stessi nomi, vengono controllati anche con l'aggiunta del suffisso ".exe".
L'inizializzazione di quest'opzione è fatta dopo aver letto il file ".vimrc" e le altre inizializzazioni, così che quando l'opzione 'shell' è stata impostata in precedenza, l'opzione 'shellredir' cambia automaticamente, a meno che sia stata impostata esplicitamente in precedenza.

In futuro le ridirezioni (pipes) possono essere usate per filtrare e
quest'opzione diverrà obsoleta (almeno per Unix).
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|,
per motivi di sicurezza.

'shellslash' *'ssl'* *'noshellslash'* *'nossl'*
'shellslash' 'ssl' booleana (default: off)
globale
{non in Vi} {solo per MSDOS, MS-Windows e OS/2}
Se è impostata, una barra (/) è usata per espandere i nomi dei file.
Ciò è utile quando una shell di tipo Unix viene usato (in ambiente
MS-DOS) al posto di command.com o di cmd.exe. I backslash possono
ancora essere utilizzati, ma sono cambiati in barre da Vim.
Nota Impostare o annullare quest'opzione non ha effetto sui nomi di
file già esistenti, e per questo motivo quest'opzione va impostata
prima di aprire qualsiasi file, per essere sicuri del risultato.
Ciò potrebbe cambiare in futuro.
'shellslash' funziona solo quando è possibile usare un backslash come
separatore nel nome di un file. Per controllare se l'opzione è
impostata, usare: >
if exists('+shellslash')
<
'shelltemp' *'stmp'* *'noshelltemp'* *'nostmp'*
'shelltemp' 'stmp' booleana (default Vi: off, default Vim: on)
globale
{non in Vi}
Quando attivata, usare file temporanei mentre si eseguono comandi
della shell. Se è inattiva, usare una "pipe".
Se non è possibile usare una "pipe", vengono usati comunque dei file
temporanei.
Al momento, una "pipe" è supportata solo in ambiente Unix e in
ambiente MS-Windows 2K e successive versione.
Potere accertarlo con: >
:if has("filterpipe")
< Il vantaggio di usare una "pipe" è che nessuno può leggere il file
temporaneo, e che il comando 'shell' non deve essere in grado di
supportare la re-direzione (degli output).
Il vantaggio nell'usare un file temporaneo è che il tipo di file e la
sua codifica possono essere accertati.
Gli autocomandi di evento |**FilterReadPre**|, |**FilterReadPost**| e
|**FilterWritePre**|, |**FilterWritePost**| non sono eseguiti quando
'shelltemp' è a off.
La funzione `system()` non tiene conto di quest'opzione e usa sempre
file temporanei.
NOTA: Quest'opzione è impostata al valore di default di Vim quando si
imposta a off 'compatible'.

'shelltype' *'st'*
'shelltype' 'st' numero (default: 0)
globale
{non in Vi} {solo per Amiga}
Su Amiga quest'opzione influenza il modo in cui funzionano i comandi
che fanno uso di una shell.
0 e 1: usare sempre la shell
2 e 3: usare la shell solo per filtrare delle righe
4 e 5: usare la shell solo per comandi ':sh'
Quando non si usa la shell, il comando è eseguito direttamente.

0 e 2: usare "shell 'shellcmdflag' cmd" per eseguire comandi esterni
1 e 3: usare "shell cmd" per eseguire comandi esterni

'shellxescape' *'sxe'*
'shellxescape' 'sxe' stringa (default: "";
per MS-DOS e MS-Windows: "\"&|<>()@^")
globale
{non in Vi}
Quando 'shellxquote' è impostato a "(", i caratteri elencati in questa
opzione saranno prefissati con un carattere '^'. Questo rende
possibile eseguire la maggior parte dei comandi esterni tramite
cmd.exe.

'shellxquote' *'sxq'*
'shellxquote' 'sxq' stringa (default: "";

per Win32, se 'shell' è cmd.exe: "("
per Win32, se 'shell' contiene
la stringa "sh" da qualche parte: "\""
per Unix, se si usa system(): "\"")
globale
{non in Vi}

Il carattere o i caratteri fra i quali viene incluso il comando passato alla shell, per i comandi "!" e ":!" . La ridirezione viene pure inclusa. Vedere 'shellquote' per escludere la ridirezione. È probabilmente inutile impostare entrambe le opzioni contemporaneamente. Quando il valore è '(' viene aggiunto ')'. Se il valore è '"(' viene aggiunto ')"' . Quando il valore è '(' vedere anche 'shellxescape'. Questa è una stringa nulla per default per la maggior parte dei sistemi, ma ne è nota l'utilità per la versione Win32, sia per cmd.exe che in automatico toglie il primo e l'ultimo apice da un comando, che per delle shell prodotte da terze parti, come la Korn Shell della MKS, o la bash, in cui dovrebbe valere "\"". Il default è adattato a seconda del valore di 'shell', per ridurre la necessità di impostare quest'opzione da parte dell'utilizzatore. Vedere |**dos-shell**|. Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'shiftround' *'sr'* *'noshiftround'* *'nosr'*
'shiftround' 'sr' booleana (default: off)
globale
{non in Vi}

Arrotondare indentatura a multipli di 'shiftwidth'. Usata per i comandi > e < . CTRL-T e CTRL-D in modo Insert arrotondano sempre l'indentatura a un multiplo di 'shiftwidth' (questo è compatibile con Vi).
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'shiftwidth' *'sw'*
'shiftwidth' 'sw' numero (default: 8)
locale al buffer

Numero di spazi da usare per ogni livello di (auto)indentatura. Usato per i comandi |**'cindent'**|, |**>>**|, |**<<**|, etc. Quando è impostata a zero verrà utilizzato il valore di 'ts'. Usare la funzione |**shiftwidth()**| per determinare il valore effettivo di shiftwidth.

'shortmess' *'shm'*
'shortmess' 'shm' stringa (default Vim: "filnxtToO", default Vi: "",
default POSIX: "A")
globale
{non in Vi}

Quest'opzione contribuisce a evitare delle richieste "Premi INVIO" |**hit-enter**| dovuti a messaggi relativi ai file, ad es. con CTRL-G, e per evitare alcuni altri messaggi.
È una lista di flag:

flag	significato se presente ~
f	usare "(3 di 5)" invece che "(file 3 di 5)"
i	usare "[noeol]" invece che "[Ultima linea incompleta]"
l	usare "999L, 888C" invece che "999 linee, 888 caratteri"
m	usare "[+]" invece che "[modificato]"
n	usare "[New]" invece che "[File Nuovo]"
r	usare "[RO]" invece che "[in sola lettura]"
w	usare "[s]" invece che "scritti" per il messaggio scrittura file e "[a]" invece che "aggiunto in fondo" per il comando ':w >> file'
x	usare "[DOS]" invece che "[in formato DOS]", "[unix]" invece che "[in formato UNIX]" e "[MAC]" invece che "[in formato MAC]"
a	tutte le abbreviazioni listate fin qui
o	sovrapponi al messaggio di scrittura file il successivo comando di lettura file (utile per ":wn" o quando 'autowrite' è attivo)
O	il messaggio di lettura file si sovrappone a qualsiasi altro messaggio precedente. Utile anche per i messaggi quickfix (ad es., ":cn").

```
            s     non inviare i messaggi "raggiunto il FONDO nella ricerca,
                  continuo dalla CIMA" e "raggiunta la CIMA nella ricerca,
                  continuo dal FONDO"
            t     troncare un messaggio di file all'inizio se è troppo lungo per
                  essere contenuto nella riga-comandi, con un simbolo "<" che
                  compare nella colonna più a sinistra.
                  Non applicato in modo Ex.
            T     troncare altri messaggi nella parte centrale se sono troppo
                  lunghi per essere contenuti nella riga-comandi, con "..."
                  al posto della parte omessa.
                  Non applicato in modo Ex.
            W     non dare messaggio "scritti" o "[s]" quando si scrive un file
            A     non date il messaggio "ATTENZIONE" quando si trova che esiste
                  già un file di swap.
            I     non dare il messaggio introduttivo alla partenza di Vim
                  |:intro|.
            c     non dare messaggi |ins-completion-menu|.  Per esempio,
                  "-- Completamento XXX (YYY)", "corrispondenza 1 di 2",
                  "L'unica corrispondenza", "Espressione non trovata",
                  "Ritorno all'originale", etc.
            q     usare "recording" invece che "recording @a"
            F     non fornire informazioni sul file, quando si sta editando
                  un file, come se si fosse usato `:silent` nel comando

        Quest'opzione può consentire evitare che, passando a un altro buffer,
        vi sia chiesto di battere <Invio>, e tuttavia dà un messaggio il più
        utile possibile, nello spazio disponibile.  Per ottenere il messaggio
        intero che avreste ricevuto impostando 'shm' alla stringa nulla, usate
        ":file!"
        Valori utili:
            shm=        Nessuna abbreviazione di messaggi.
            shm=a       Abbreviazione, senza perdita di informazione.
            shm=at      Abbreviazione, e troncamento del messaggio se
                        necessario.

        NOTA: Quest'opzione è impostata al valore di default di Vi
        impostando 'compatible' e al valore di default di Vim quando
        'compatible' viene messa a off.

                                    *'shortname'* *'sn'* *'noshortname'* *'nosn'*
'shortname' 'sn'        booleana        (default: off)
                        locale al buffer
                        {non in Vi, non nelle versioni MS-DOS}
        Si suppone che i nomi di file siano lunghi al massimo 8 caratteri, più
        un suffisso di 3 caratteri.  Non è permesso che un solo "." nel nome
        del file.  Quando quest'opzione è attivata, i "." nei nomi di file
        sono rimpiazzati con caratteri "_", quando serva aggiungere
        un'estensione (".~" o ".swp").  Quest'opzione non è disponibile
        per MS-DOS, perché in quel caso sarebbe sempre attiva.  Quest'opzione
        è utile quando si editano dei file in un File System che sia
        compatibile con MS-DOS, ad es., messydos o crossdos.  Quando si
        esegue la versione GUI Win32 sotto Win32s, quest'opzione è sempre
        attiva per default.

                                                *'showbreak'* *'sbr'* *E595*
'showbreak' 'sbr'       stringa (default: "")
                        globale
                        {non in Vi}
                        {non disponibile se compilato senza la funzionalità
                        |+linebreak|}
        Stringa da visualizzare all'inizio di righe che siano state
        "spezzate", (ossia che sono visualizzate in più di una riga dello
        schermo).  Valori utili sono "> " o "+++ ". >
                :set showbreak=>\
<       Nota Il "\" serve per proteggere lo spazio finale.  È più semplice
        così: >
                :let &showbreak = '+++ '
<       Solo caratteri stampabili e che occupino un solo byte sullo schermo
        sono consentiti, tranne <Tab> e "," (in una futura versione la ","
        potrebbe essere usata per separare la parte visualizzata all'inizio e
        alla fine di una riga sullo schermo).
        Il caratteri sono evidenziati come specificato dal flag '@'
        dell'opzione 'highlight'.
```

Nota I caratteri <Tab> dopo il carattere di 'showbreak' saranno visualizzati in maniera differente. Se volete che 'showbreak' sia posto all'interno della numerazione delle righe, aggiungete a 'cpoptions' il flag "n".

'showcmd' *'sc'* *'noshowcmd'* *'nosc'*
'showcmd' 'sc' booleana (Vim default: on, off per Unix, Vi default: off, impostata a on in |**defaults.vim**|)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+cmdline_info**|}

Visualizza i comandi (incompleti) nell'ultima riga dello schermo. Impostate quest'opzione a off se state lavorando su un terminale lento.
In modo Visual viene visualizzata la dimensione dell'area selezionata:
- Quando si selezionano caratteri all'interno di una riga, il numero dei caratteri. Se il numero di byte è differente, anche questo viene visualizzato: "2-6" sta a significare 2 caratteri e 6 byte. (almeno uno dei caratteri è solitamente un <Tab>).
- Quando si seleziona più di una riga, il numero delle righe.
- Quando si seleziona un blocco di testo, la dimensione è espressa in numero di caratteri sullo schermo: {righe}x{colonne}.

NOTA: Quest'opzione è impostata al valore di default di Vi impostando 'compatible' e al valore di default di Vim quando 'compatible' viene messa a off.

'showfulltag' *'sft'* *'noshowfulltag'* *'nosft'*
'showfulltag' 'sft' booleana (default: off)
globale
{non in Vi}

Quando si completa una parola in modo Insert (vedere |**ins-completion**|) dal file dei tag, visualizzare sia il nome del tag che una forma "pulita" della stringa usata per la ricerca (se ce n'è una), come possibili corrispondenze. Quindi se la corrispondenza cercata è quella di una funzione C, si può vedere un modello degli argomenti che sono richiesti dalla funzione (se la cosa è possibile, in quel particolare stile di codifica).
Nota Questo non funziona bene se nell'opzione 'completeopt' viene specificato "longest" (la più lunga possibile), perché ciò che viene trovato dalla ricerca dell'espressione può non corrispondere al testo che era stato battuto (ossia si trova un'espressione "equivalente", ma più "corta").

'showmatch' *'sm'* *'noshowmatch'* *'nosm'*
'showmatch' 'sm' booleana (default: off)
globale

Quando si inserisce una parentesi, saltare per un momento alla parentesi corrispondente. Il salto viene effettuato solo se la corrispondenza è visibile sullo schermo. Il tempo in cui viene mostrata la corrispondenza può essere impostato con 'matchtime'.
Viene dato un Bip se la corrispondenza non esiste (tanto nel caso che la corrispondenza sia visibile, come nel caso in cui sia "fuori dallo schermo".
Quest'opzione è messa a off quando l'opzione 'paste' viene impostata e ripristinata quando 'paste' è messa a off.
Quando il flag 'm' non è incluso in 'cpoptions', immettendo un carattere il cursore torna subito nella posizione appropriata.
Vedere il campo "sm" in 'guicursor' per impostare la forma del cursore e il lampeggiamento mentre si mostra la corrispondenza.
L'opzione 'matchpairs' può essere usata per specificare i caratteri per i quali mostrare le corrispondenze. Le opzioni 'rightleft' e 'revins' sono usate per cercare corrispondenze opposte.
Vedere anche il "plugin" matchparen per evidenziale la corrispondenza mentre ci si sposta nel file |**pi_paren.txt**|.

'showmode' *'smd'* *'noshowmode'* *'nosmd'*
'showmode' 'smd' booleana (default Vim: on, default Vi: off)
globale

Se in modo Insert, Replace o Visual segnalalo con un messaggio sull'ultima riga dello schermo.
Usate il flag 'M' in 'highlight' per impostare il tipo di evidenziazione da usare per questo messaggio.

Quando si può usare|**XIM**|il messaggio dirà anche "XIM". Ma questo non vuol dire che XIM è veramente attivo, in particolare quando 'imactivatekey' non è impostato.
NOTA: Quest'opzione è impostata al valore di default di Vi impostando 'compatible' e al valore di default di Vim quando 'compatible' viene messa a off.

'showtabline' *'stal'*
'showtabline' 'stal' numero (default: 1)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+windows**|}
Il valore di quest'opzione specifica quando la riga con le descrizioni relative alle linguette sarà visualizzata:
0: mai
1: solo se ci sono almeno due linguette
2: sempre
Questo vale sia per l'implementazione GUI che per quella non-GUI della riga delle descrizioni delle linguette.
Vedere |**tab-page**| per un'informazione ulteriore riguardo alle linguette.

'sidescroll' *'ss'*
'sidescroll' 'ss' numero (default: 0)
globale
{non in Vi}
Il numero minimo di colonne da scorrere orizzontalmente. In uso solo quando l'opzione 'wrap' è a off e il cursore viene spostato sullo schermo.
Quando è a zero il cursore sarà posto in mezzo allo schermo. Se si usa un terminale lento, impostatelo a un numero alto o a 0. Quando si usa un terminale veloce, usare un numero piccolo, o 0. Non usato per i comandi "zh" e "zl".

'sidescrolloff' *'siso'*
'sidescrolloff' 'siso' numero (default: 0)
globale
{non in Vi}
Il numero minimo di colonne dello schermo da mantenere alla sinistra e alla destra del cursore se l'opzione 'nowrap' è attivata. Impostare quest'opzione a un valore maggiore di 0 mentre |'**sidescroll**'| è pure impostata a un valore diverso da zero rende visibile qualche contesto nella riga nella quale state scorrendo orizzontalmente (tranne che all'inizio e alla fine della riga). Impostare quest'opzione a un valore elevato (come 999) ha come effetto di mantenere il cursore centrato orizzontalmente nella finestra, almeno finché non ci si avvicina troppo all'inizio o alla fine della riga.
NOTA: Quest'opzione è impostata a 0 se si attiva 'compatible'.

Esempio: Provate queste impostazioni insieme a 'sidescroll' e 'listchars' come nell'esempio seguente, per impedire che il cursore si metta sui caratteri "di estensione": >

```
:set nowrap sidescroll=1 listchars=extends:>,precedes:<
:set sidescrolloff=1
<
```

'signcolumn' *'scl'*
'signcolumn' 'scl' stringa (default: "auto")
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+signs**|}
Se visualizzare o no la colonna del segno. I valori che si possono specificare sono:
"auto" solo se c'è un segno da visualizzare
"no" mai
"yes" sempre

'smartcase' *'scs'* *'nosmartcase'* *'noscs'*
'smartcase' 'scs' booleana (default: off)
globale
{non in Vi}

Ignora l'opzione 'ignorecase' se l'espressione da cercare contiene lettere maiuscole. Usata solo quando l'espressione da cercare viene specificata al momento e l'opzione 'ignorecase' è attiva. Usata per i comandi "/", "?", "n", "N", ":g" e ":s". Non usata per "*", "#", "gd", ricerca di tag, etc. Dopo "*" e "#" potete attivare 'smartcase' battendo un comando "/", richiamando l'espressione da cercare dalla lista "storica" e battendo <Invio>.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'smartindent' *'si'* *'nosmartindent'* *'nosi'*
'smartindent' 'si' booleana (default: off)
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità |+smartindent|}
Serve a indentare automaticamente in maniera intelligente all'inizio di una nuova riga. Funziona per programmi della famiglia del C, ma può essere usata anche per altri linguaggi. 'cindent' fa qualcosa di simile, funziona meglio nella maggior parte dei casi, ma è più rigida, vedere |**C-indenting**|. Quando 'cindent' è attiva o 'indentexpr' è sgtata specificata, impostare 'si' non produce alcun effetto. 'indentexpr' è un'alternativa più evoluta.
Normalmente anche 'autoindent' dovrebbe essere attiva quando si usa 'smartindent'.
Un'indentatura è inserita automaticamente;
- Dopo una riga che finisce con '{'.
- Dopo una riga che inizia con una espressione contenuta in 'cinwords'.
- Prima di una riga che inizia con '}' (solo con il comando "O").
Quando si batte '}' come primo carattere di una nuova riga, a quella riga è data la stessa indentatura della corrispondente '{'.
Quando si batte '#' come primo carattere di una nuova riga, l'indentatura di quella riga viene rimossa, il carattere '#' è messo nella prima colonna. L'indentatura è ripristinata alla successiva riga. Se non desiderate ciò, usate la seguente mappatura: ":inoremap # X^H#", dove ^H è immesso battendo CTRL-V CTRL-H.
Quando si usa il comando ">>", le righe che iniziano con '#' non sono spostate a sinistra.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.
Quest'opzione è messa a off quando si attiva l'opzione 'paste' e ripristinata quando l'opzione 'paste' è messa a off.

'smarttab' *'sta'* *'nosmarttab'* *'nosta'*
'smarttab' 'sta' booleana (default: off)
globale
{non in Vi}
Se impostata, una <Tab> a inizio riga inserisce tanti spazi bianchi quanti specificati in 'shiftwidth'. 'tabstop' o 'softtabstop' sono usati in altre posizioni. Un <BS> cancellerà il numero di spazi bianchi specificato in 'shiftwidth' a inizio riga.
Se inattiva, una <Tab> inserisce sempre spazi bianchi nel numero necessario per soddisfare le specifiche di 'tabstop' e 'softtabstop'. 'shiftwidth' è usato solo per spostare testo verso sinistra o verso destra, vedere |**shift-left-right**|.
Quel che viene inserito (una <Tab> o degli spazi) dipende dall'opzione 'expandtab'. Vedere anche |**ins-expandtab**|. Quando 'expandtab' non è impostata, il numero di spazi viene minimizzato usando delle <Tab>.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.
Quest'opzione è messa a off quando si imposta l'opzione 'paste' e ripristinata quando l'opzione 'paste' è messa a off.

'softtabstop' *'sts'*
'softtabstop' 'sts' numero (default: 0)
locale al buffer
{non in Vi}
Numero di spazi che una <Tab> vale mentre si eseguono operazioni di modifica, come l'inserimento di una <Tab> o l'uso del tasto <BS>. "Sembra" di inserire delle <Tab> mentre in effetti si inserisce una stringa mista di spazi e di <Tab>. Ciò è utile se si vuol mantenere l'opzione 'ts' al suo valore standard, che è 8, mentre durante la modifica è impostato a 'sts'. Comunque, comandi come "x" lavorano sempre sul carattere effettivamente esistente nel testo.
Quando 'sts' è a zero, il comportamento è quello standard.

Quando 'sts' ha un valore negativo, viene usato il valore di
'shiftwidth'.
'softtabstop' è messa a zero quando l'opzione 'paste' è impostata,
e rispristinata quando 'paste' è messo a off.
Vedere anche |**ins-expandtab**|. Quando 'expandtab' non è impostato,
il numero di spazi inserito è minimizzato usando delle <Tab>.
il flag 'L' in 'cpoptions' cambia il modo di usare <Tab> quando si
imposta 'list'.
NOTA: Quest'opzione è impostata a 0 se si attiva 'compatible'.

'spell' *'nospell'*

'spell' booleana (default: off)
locala alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+syntax**|}

Quanto è attivata, verrà effettuata la correzione ortografica.
Vedere |**spell**|.
Le lingua sono specificate con 'spelllang'.

'spellcapcheck' *'spc'*

'spellcapcheck' 'spc' stringa (default: "[.?!]_[\])'" \t]\+")
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+syntax**|}

Espressione regolare per identificare la fine di una frase. La parola
che viene dopo ognuno di questi caratteri verrà controllata, perché
dovrebbe iniziare con la lettera maiuscola. Se non è così, la parola
viene evidenziata con SpellCap |**hl-SpellCap**| (a meno che la parola non
sia anche scritta male).
Quando non si desidera questo controllo, assegnate al valore di
quest'opzione la stringa nulla.
L'opzione è usata solo se 'spell' è attivato.
Fare attenzione ai caratteri speciali, vedere |**option-backslash**| per
includere spazi e backslash.
Per impostare quest'opzione automaticamente a seconda della lingua,
vedere |**set-spc-auto**|.

'spellfile' *'spf'*

'spellfile' 'spf' stringa (default: "")
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+syntax**|}

Nome del file che contiene la lista di parole (wordlist) dove delle
parole nuove vengono aggiunte con i comandi |**zg**| e |**zw**|. Deve
terminare in ".{encoding}.add". Va specificato il come completo di
percorso, altrimenti il file viene posto nella directory corrente.
E765
Può anche essere una lista di nomi, separati da virgole. Un contatore
prima dei comandi |**zg**| e |**zw**| può essere usato per accedere a ognuna di
esse. Ciò consente di usare una lista di parole personale, assieme a
una lista di parole di progetto.
Quando una parola viene aggiunta mentre quest'opzione non è
specificata, Vim la imposterà per voi: sarà usata la prima directory
su cui si possa scrivere in 'runtimepath'. Se non esiste ancora un
directory "spell", verrà creata. Come nome di file si userà il primo
nome di lingua che appare in 'spelllang', ignorando le eventuali
varianti regionali.
Il file risultante ".spl" sarà usato per la correzione ortografica,
non è necessario che compaia in 'spelllang'.
Normalmente si usa un file per tutte le regioni, ma è possibile
aggiungere il nome della regione, se si preferisce. Comunque, verrà
usato solo quando 'spellfile' è impostato col suo nome, perché
verranno trovati elementi in file di 'spelllang' senza nome della
regione.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|,
per motivi di sicurezza.

'spelllang' *'spl'*

'spelllang' 'spl' stringa (default: "en")
locale al buffer

{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+syntax**|}
Una lista, separata da virgole, di nomi di liste di parole (wordlist).
Quando l'opzione 'spell' è attiva, la correzione ortografica sarà
effettuata per queste lingue. Esempio: >
set spelllang=en_us,nl,medical
< Ciò vuol dire che le parole in inglese (USA), in olandese e termini
medici saranno riconosciute. Le parole che non sono riconosciute
verranno evidenziate.
Il nome della lista di parole non deve includere una virgola o un
punto. L'uso di un trattino è raccomandato per separare
l'identificativo di due lettere della lingua da una specificazione
Quindi "en-rare" è usato per parole inglesi raramente usate.
Un nome di regione deve venire per ultimi e avere la forma "_xx",
dove "xx" è il nome di due lettere minuscole, della regione.
Si possono usare più regioni, elencandole: "en_us,en_ca" supporta
sia l'inglese USa che quello canadese, ma non le parole specifiche
ad Australia, Nuova Zelanda o Gran Bretagna. (Nota: al momento
le liste di parole en_au ed en_nz sono più datate di quelle
relative a en_ca, en_gb ed en_us).
Se il nome "cjk" è incluso, i caratteri in uso in Estremo Oriente
sono esclusi dalla correzione ortografica. Questo è utile quando si
modifichino testi che contengono parole in lingue asiatiche.
E757
Come caso speciale, il nome di un file .spl può essere fornito come
percorso completo. Il primo "_xx" nel nome viene tolto e usato come
nome della regione (_xx è un carattere di sottolineatura, seguito da
due lettere, e a sua volta seguito da qualcosa che non è una lettera).
Ciò è principalmente da utilizzare per dei test. Dovete assicurarvi
che la codifica corretta sia usata nel file, Vim non la controlla.
Quando 'encoding' è impostato, le liste di parole sono nuovamente
caricate. Quindi è una buona idea impostare 'spelllang' dopo aver
impostato 'encoding' per evitare di caricare i file due volte.
Come si trovano i file usati per la correzione (file di spell) è
spiegato qui: |**spell-load**|.

Se il plugin |**spellfile.vim**| è attivo e usare un linguaggio per il
quale Vim non trova il corrispondente file .spl in 'runtimepath' il
plugin richiederà se volete scaricare il file da Internet.

Dopo che quest'opzione è stata impostata con successo, Vim eseguirà i
file "spell/LANG.vim" in 'runtimepath'. "LANG" è il valore di
'spelllang' fino alla prima virgola, punto o sottolineatura.
Vedere anche |**set-spc-auto**|.

'spellsuggest' *'sps'*
'spellsuggest' 'sps' stringa (default: "best")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+syntax**|}
Metodi usati per suggerire una correzione. Valgono sia per il
comando |**z=**| che per la funzione |**spellsuggest()**|. Si tratta di una
lista di elementi, separati da virgola:

best Metodo interno, che funziona meglio per l'inglese.
Trova le possibili alternative come il metodo "fast"
e usa in piccola parte un punteggio "suono-simile-a"
per migliorare l'ordine in cui le alternative sono
elencate.

double Metodo interno, che usa due metodi, e fonde i
risultati. Il primo metodo è "fast", l'altro metodo
calcola quando la parola suggerita ha una pronuncia
simile a quella della parola sbagliata. Ciò vale solo
se la lingua specifica "sound-folding" (qualcosa tipo
"piegatura del suono"). Può essere lento, e non sempre
dà i risultati migliori.

fast Metodo interno, che controlla solo per modifiche
semplici: inserimento/omissione/scambio di caratteri.

Funzione bene per gli errori di battitura più comuni.

{number} Il numero massimo di suggerimenti elencati con il comando |**z=**|.
Non usato dalla funzione |**spellsuggest()**|.
Il numero dei suggerimenti non è mai superiore al valore di 'lines' meno due.

file:{nomefile} Usare il file {nomefile},che deve consistere in righe di due colonne, separate da una "/". La prima colonna contiene la parola sbagliata, la seconda colonna contiene la parola giusta da suggerire.
Esempio:

```
theribal/terrible ~
```

Da usare per gli errori frequenti che non figurano in cima alla lista dei suggerimenti, utilizzando i metodi interni. Le righe senza una "/" vengono ignorate, si possono usare per inserire dei commenti.
La parola nella seconda colonna deve essere esatta, altrimenti non verrà utilizzata. Aggiungete la parola al file ".add" se la parola stessa è attualmente ritenuta un errore ortografico.
Il file è usato per tutte le lingue.

expr:{espressione} Valutare l'espressione {espressione}. Usare una funzione per evitare problemi con gli spazi. |**v:val**| contiene la parola sbagliata. L'espressione deve assumere come valore una Lista di Liste (List of Lists), ognuna della quali contiene un suggerimento e un punteggio.
Esempio:

```
[['the', 33], ['that', 44]] ~
```

Impostare 'verbose' e usare |**z=**| per vedere i punteggi che sono usati dai metodi interni. Un punteggio più basso è migliore di uno più alto.
Questo metodo può invocare |**spellsuggest()**| se si imposta temporaneamente 'spellsuggest' in modo da escludere la parte "expr:".
Gli errori sono ignorati senza alcun messaggio, a meno che impostiate l'opzione 'verbose' a un valore diverso da zero.

È consentito usare solo uno dei metodi "best", "double" o "fast". Gli altri parametri possono comparire anche più volte in un ordine qualsiasi. Esempio: >

```
:set sps=file:~/.vim/sugg,best,expr:MySuggest()
```

<

Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'splitbelow' *'sb'* *'nosplitbelow'* *'nosb'*
'splitbelow' 'sb' booleana (default: off)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+windows**|}

Se impostata, dividendo in due una finestra, la nuova finestra verrà messa sotto la finestra corrente. Vedere |**:split**|.

'splitright' *'spr'* *'nosplitright'* *'nospr'*
'splitright' 'spr' booleana (default: off)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+vertsplit**|}

Se impostata, dividendo in due una finestra, la nuova finestra verrà messa alla destra della finestra corrente. Vedere |**:vsplit**|.

'startofline' *'sol'* *'nostartofline'* *'nosol'*
'startofline' 'sol' booleana (default: on)
globale
{non in Vi}

Quando impostata a on i comandi elencati sotto spostano il cursore
al primo spazio non-bianco della riga. Quando l'opzione è a off il
cursore è mantenuto nella stessa colonna (se possibile). Questo vale
per i comandi: CTRL-D, CTRL-U, CTRL-B, CTRL-F, "G", "H", "M", "L", gg,
e per i comandi "d", "<<" e ">>", con un operatore che agisce sulla
riga, con "%" con un contatore, e a comandi che cambiano di buffer
(CTRL-^, :bnext, :bNext, etc.). Anche per un comando di Ex che abbia
un solo numero di riga, ad es., ":25" o ":+".
Nel caso di comandi che cambiano di buffer, il cursore è messo sulla
colonna in cui si trovava l'ultima volta che quel buffer era stato
editato.
NOTA: Quest'opzione è impostata a on se si attiva 'compatible'.

'statusline' *'stl'* *E540* *E542*
'statusline' 'stl' stringa (default: "")
globale o locale alla finestra |**global-local**|
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+status_line**|}
Se specificata, quest'opzione determina il contenuto della riga di
status. Vedere anche |**status-line**|.

L'opzione consiste di elementi in stile "printf" ('%') e di parti di
testo normale. Ogni elementi della riga di status ha la forma:
%-0{largmin}.{largmax}{elemento}
Tutti i campi, tranne {elemento} sono opzionali. Per inserire un
simbolo di "%", immettere "%%". È possibile specificare fino a 80
elementi. *E541*

Quando l'opzione inizia con "%!" viene trattata come un'espressione,
che viene valutata, e il risultato è usato come valore dell'opzione.
Esempio: >
:set statusline=%!MyStatusLine()
< Il risultato può contenere elementi %{} che verranno a loro volta
valutati.
Notare che l'espressione "%!" è valorizzata nel contesto della
finestra e del buffer corrente, mentre gli elementi %{} sono
valorizzati nel contesto della finestra a cui la statusline
appartiene.

Nel caso si riscontri un errore nella valorizzazione dell'opzione,
all'opzione stessa verrà assegnato il valore nullo, per evitare errori
ulteriori. In caso contrario l'aggiornamento dello schermo verrebbe
eseguito in continuazione [generando un ciclo infinito].

Nota Il solo effetto dell'opzione 'ruler' quando quest'opzione è
impostata (e 'laststatus' vale 2) è quello di specificare l'output
del comando |**CTRL-G**|.

campo	significato ~
-	Giustificare a sinistra l'elemento. Per default viene giustificato a destra quando "largmin" è maggiore della lunghezza dell'elemento.
0	Premettere degli zeri se l'elemento è un numero. Non ha effetto se si specifica '-'.
largmin	Larghezza minima dell'elemento, con riempimento impostato da '-' & '0'. Il valore deve essere inferiore a 50.
largmax	Larghezza massima dell'elemento. Un troncamento viene effettuato con un simbolo '<' a sinistra, per elementi di tipo testo. Elementi numerici sono ridotti a un numero di cifre "maxwid"-2, seguite da un simbolo ">" e da un numero, che dice quante cifre sono omesse, con una notazione simile a quella esponenziale.
elemento	Una codifica di una lettera, come descritto più sotto.

Segue una descrizione degli elementi che possono essere presenti in
una riga di status. Il secondo carattere di "elemento" è il tipo:
N per numero
S per stringa
F per flag come descritta sotto
- non consentito

elem. significato e formato ~
f S Percorso del file nel buffer, come immesso, oppure relativo alla directory corrente.
F S Nome completo del file nel buffer.
t S Coda (tail) ossia parte finale del nome del file nel buffer.
m F Flag file modificato, "[+]"; "[-]" se 'modifiable' è a off.
M F Flag file modificato, ",+" o ",-".
r F Flag di file in sola lettura, "[RO]".
R F Flag di file in sola lettura, ",RO".
h F Flag di buffer di help, "[help]".
H F Flag di buffer di help, ",HLP".
w F Flag di finestra di anteprima (preview), "[Preview]".
W F Flag di finestra di anteprima (preview), ",PRV".
y F Tipo di file nel buffer, ad es., "[vim]". Vedere 'filetype'.
Y F Tipo di file nel buffer, ad es., ",VIM". Vedere 'filetype'.
q S "[Quickfix List]", "[Location List]" oppure vuoto.
k S Valore di "b:keymap_name" o 'keymap' quando mappature |:lmap| sono in uso: "<keymap>"
n N Numero buffer.
b N Valore del carattere sotto il cursore.
B N Come sopra, in esadecimale.
o N Numero di byte, nel file, del byte sotto il cursore, il primo byte ha il numero 1. La "o" sta per "offset", questo è l'offset dall'inizio del file (con l'aggiunta di uno).
{non disponibile se compilato senza la funzionalità |+byte_offset|}
O N Come sopra, in esadecimale.
N N Numero pagina di stampa. (Funziona solo nell'opzione 'printheader')
l N Numero di riga.
L N Numero di righe nel buffer.
c N Numero di colonna.
v N Numero di colonna virtuale [espandendo i <TAB>].
V N Numero di colonna virtuale in formato -{num}. Non visualizzato se uguale all'elemento 'c'.
p N Percentuale all'interno del file in righe, come usando |CTRL-G|.
P S Percentuale all'interno del file della finestra visualizzata. Questo è simile alla percentuale descritta per 'ruler'. Sempre lunga 3, a meno che non sia stata tradotta.
a S Status della lista argomenti, come nel titolo della finestra di titolo di default. ({numerobuffercorrente} di {numeromassimo}) Nullo se i file specificati sono zero o uno.
{ NF Valutare l'espressione fra '%{' e '}' e sostituire il risultato. Nota Non c'è un "%" prima della chiusa graffa "}". L'espressione non può contenere un carattere '}', occorre chiamare una funzione laddove questo fosse necessario.
(- Inizio di un gruppo di elementi. Si può usare per impostare la larghezza e l'allineamento di una sezione. Deve essere seguito da %) prima o poi.
) - Fine di un gruppo di elementi. Non è possibile specificare delle larghezze.
T N Per 'tabline': inizio etichetta della linguetta N. Usare %T dopo l'ultima etichetta. Quest'informazione è usata per i clic del mouse.
X N per 'tabline': inizio della fine etichetta della linguetta N. Usare %X dopo l'etichetta, ad es.: %3Xclose%X. Uare %999X per marcare la chiusura della linguetta corrente (close current tab) Quest'informazione è usata per i clic del mouse.
< - Dove troncare la riga se diventa troppo lunga. Per default la si tronca all'inizio. Non è possibile specificare delle larghezze.
= - Punto di separazione fra elementi allineati a sinistra e a destra. Non è possibile specificare delle larghezze.
- Imposta gruppo evidenziazione. Il nome segue e poi si mette ancora un #. Quindi usare %#HLnome# per indicare il gruppo di evidenziazione HLnome. Lo stesso tipo di evidenziazione è usato anche per la riga di status delle finestre diverse da quella corrente.
* - Impostare gruppo di evidenziazione a User{N}, dove {N} è preso dal campo "larghmin". Ad es. %1*. Ripristinare evidenziazione normale con %* o %0*.
La differenza fra User{N} e StatusLine sarà applicata a StatusLineNC per la riga di status delle finestre che non sono

quella corrente. Il numero deve essere fra 1 e 9.
Vedere |**hl-User1..9**|.

Quando si visualizza un flag, Vim omette la virgola che lo precede, se ce n'è una, quando quel flag viene subito dopo del testo. Ciò rende la visualizzazione più armoniosa quando i flag sono usati come negli esempi più sotto.

Quando tutti gli elementi che fanno parte di un gruppo si riducono a una stringa nulla (ossia si tratta di flag non impostate), e una "largmin" non è impostata per il gruppo, l'intero gruppo si svuota. Questo farà sì che un gruppo come quello che segue sparisca dalla riga di status quando nessuno dei flag è impostato. >
```
:set statusline=...%(\ [%M%R%H]%)...
```
<
g:actual_curbuf
Prestare attenzione al fatto che un'espressione viene valutata ogniqualvolta la riga di status viene visualizzata. Il buffer corrente e la finestra corrente saranno temporaneamente impostati a quello della finestra (e del buffer) la cui riga di status è in fase di visualizzazione. L'espressione verrà valutata in questo contesto. La variabile "actual_curbuf" è impostato a 'bufnr()', ovvero al numero del buffer corrente effettivo.

L'opzione 'statusline' sarà valutata nel |**sandbox**| se impostata da una modeline; vedere |**sandbox-option**|.

Non è consentito cambiare testo o passare a un'altra finestra mentre è in corso la valutazione di 'statusline' |**textlock**|.

Se la riga di status non viene aggiornata quando lo desiderate (ad es., dopo aver impostato una variabile che è usata in un'espressione, potete forzare un aggiornamento impostando una variabile senza alterarne il valore. Ad es.: >
```
:let &ro = &ro
```
<
Un risultato che contiene solo cifre è considerato un numero, agli effetti della visualizzazione.
In caso contrario, il risultato è considerato testo di flag e vengono applicate le regole descritte sopra.

Fare attenzione agli errori nelle espressioni. Possono rendere Vim inutilizzabile!
Se vi trovate bloccati, tenete premuto ':' o 'Q' per poter immettere comandi, poi uscite dall'editor per modificare il vostro .vimrc o quel che va modificato usando "vim --clean" per poterlo fare senza problemi.

Esempi:
Emulare la riga di status standard, quando 'ruler' è impostato >
```
:set statusline=%<%f\ %h%m%r%=%-14.(%l,%c%V%)\ %P
```
<
Simile, ma aggiungere il valore ASCII del carattere sotto il cursore (come "ga") >
```
:set statusline=%<%f%h%m%r%=%b\ 0x%B\ \ %l,%c%V\ %P
```
<
Visualizza contatore byte e valore del byte, flag "modificato" in rosso. >
```
:set statusline=%<%f%=\ [%1*%M%*%n%R%H]\ %-19(%3l,%02c%03V%)%O'%02b'
:hi User1 term=inverse,bold cterm=inverse,bold ctermfg=red
```
<
Visualizza un flag ,GZ se si sta caricando un file compresso >
```
:set statusline=...%r%{VarExists('b:gzflag','\ [GZ]')}%h...
```
<
In |**:autocmd**| c'è: >
```
:let b:gzflag = 1
```
<
E: >
```
:unlet b:gzflag
```
<
E definire questa funzione: >
```
:function VarExists(var, val)
:    if exists(a:var) | return a:val | else | return '' | endif
:endfunction
```
<
'suffixes' *'su'*
'suffixes' 'su' stringa (default: ".bak,~,.o,.h,.info,.swp,.obj")
globale
{non in Vi}
File che hanno questi suffissi avranno una priorità inferiore quando

un'espressione di ricerca corrisponde a più di un file. Vedere |**suffixes**|. Virgole si possono usare per separare i suffissi. Spazi dopo la virgola sono ignorato. Un punto '.' è visto anche come l'inizio di un suffisso. Per evitare che un punto o una virgola vengano trattati come separatori, fateli precedere da un backslash. Vedere |**option-backslash**| per inserire spazi e backslash. Vedere 'wildignore' per ignorare del tutto dei file. L'uso di |**:set+=**| e |**:set-=**| è da preferire quando si aggiungono o si tolgono suffissi dalla lista. Ciò permette di evitare problemi laddove una futura versione usasse un altro default.

'suffixesadd' *'sua'*

'suffixesadd' 'sua' stringa (default: "")
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+file_in_path**|}

lista di suffissi, separati da virgole, che sono usati nella ricerca di un file, tramite i comandi "gf", "[I", etc. Ad es.: >

```
:set suffixesadd=.java
```

<

'swapfile' *'swf'* *'noswapfile'* *'noswf'*

'swapfile' 'swf' booleana (default: on)
locale al buffer
{non in Vi}

Usare un file di swap come buffer. Quest'opzione può essere annullata se non si desidera usare un file di swap per un particolare buffer. Ad esempio, per informazioni confidenziale cui neppure l'utente "root" deve essere in grado di accedere.
Attenzione: Tutto il testo sarà solo in memoria:
- Non usare per grossi file.
- Il ripristino in caso di problemi sarà impossibile!

Un file di swap sarà present solo quando |**'updatecount'**| è diverso da zero e 'swapfile' è stato impostato.
Quando 'swapfile' è annullato, il file di swap per il buffer corrente è immediatamente cancellato. Quando 'swapfile' è impostato, e 'updatecount' è diverso da zero, un file di swap è immediatamente creato.
Vedere anche |**swap-file**| e |**'swapsync'**|.
Per aprire un nuovo buffer senza che venga creato anche un file di swap, usate il modificatore |**:noswapfile**|.
Vedere 'directory' per il posizionamento del file di swap.

Quest'opzione è usata insieme a 'bufhidden' e 'buftype' per specificare tipi speciali di buffer. Vedere |**special-buffers**|.

'swapsync' *'sws'*

'swapsync' 'sws' stringa (default: "fsync")
globale
{non in Vi}

Quando quest'opzione non è nulla un file di swap viene sincronizzato su disco dopo averlo scritto. Questo richiede del tempo, specie su sistemi Unix altamente utilizzati. Quando quest'opzione è nulla parti del file di swap possono trovarsi in memoria, e non essere ancora stati scritti su disco. Se il sistema si guasta, potreste perdere una parte maggiore del vostro lavoro. In Unix il sistema esegue un comando "sync" ogni tanto, anche senza che Vim lo richieda esplicitamente, quindi lo svantaggio nel lasciare nulla quest'opzione è piccolo. In alcuni sistemi il file di swap non verrà scritto per nulla [in quanto non necessario - NdT]. Per un sistema Unix, impostare quest'opzione a "sync" richiederà di usare la chiamata di sistema sync() invece di fsync(), che è il valore di default, e questo può dare risultati migliori per alcuni sistemi.
l'opzione 'fsync' è usata per il file vero e proprio.

'switchbuf' *'swb'*

'switchbuf' 'swb' stringa (default: "")
globale
{non in Vi}

Quest'opzione controlla il comportamento quando si passa da un buffer a un altro.
Valori possibili (lista separata da virgole):
useopen Se incluso, usare la prima finestra aperta che

contiene il buffer specificato (se ce n'è uno).
Altrimenti: Non considerare le altre finestre.
Questa impostazione è controllata con comandi
|**quickfix**|, quando si salti a righe contenenti errori.
(":cc", ":cn", "cp", etc.). È anche usato in tutti i
comandi di split [creazione nuova finestra] relativi a
buffer, per esempio ":sbuffer", ":sbnext", o
":sbrewind".

usetab Come "useopen", ma prendi in considerazione anche finestre in altre linguette.

split Se incluso, dividere in due la finestra corrente prima di caricare un buffer per un comando |**quickfix**| che visualizzi gli errori. Altrimenti: non dividere in due, ma usare la finestra corrente.

vsplit Come "split", ma divide verticalmente.

newtab Come "split", ma apre una nuova linguetta. Prevale su "split" se sono entrambi presenti.

'synmaxcol' *'smc'*

'synmaxcol' 'smc' numero (default: 3000)
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+syntax**|}

Massima colonna in cui ricercare elementi di sintassi. Per righe lunghe, il testo oltre questa colonna non viene evidenziato e le righe seguenti possono non essere evidenziate correttamente, perché lo "status" dell'evidenziazione sintattica è ri-inizializzato.
Questo può servire a evitare un rinfresco molto lento della videata per un file XML che consista di un'unica lunga riga.
Impostare a zero per rimuovere il limite.

'syntax' *'syn'*

'syntax' 'syn' stringa (default: "")
locale al buffer
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+syntax**|}

Quando quest'opzione è impostata, il file sintattico con il nome indicato viene caricato, tranne nel caso in cui l'evidenziazione sintattica sia stata inibita col comando ":syntax off".
Altrimenti quest'opzione non sempre riflette la sintassi corrente (la variabile b:current_syntax è invece accurata).
Quest'opzione è molto utile in una modeline, per un file la cui sintassi non sia riconosciuta automaticamente. Ad es., in un file IDL:

```
/* vim: set syntax=idl : */ ~
```

Quando un "." appare nel valore, esso serve a delimitare due nomi di tipi di file. Esempio:

```
/* vim: set syntax=c.doxygen : */ ~
```

In questo caso verrà usata dapprima la sintassi "c", e poi quella di "doxygen" syntax.
Nota L'ulteriore sintassi specificata dev'essere disponibile per poterla caricare in addizione all'altra, altrimenti verrà saltata.
Può esserci più di un ".".
Per inibire l'evidenziazione sintattica per il file corrente, usare: >

```
:set syntax=off
```

< Per attivare l'evidenziazione sintattica usando il valore corrente dell'opzione 'filetype': >

```
:set syntax=on
```

< Quel che veramente succede quando si imposta l'opzione 'syntax' è che viene attivato l'autocomando Syntax, con il valore dell'opzione come argomento.
Quest'opzione non è copiata in un altro buffer, a prescindere dalle flag 's' o 'S' in 'cpoptions'.
Solo caratteri normali possono essere usati nei nomi di file, i caratteri "/*?[|<>" non sono validi.

'tabline' *'tal'*

'tabline' 'tal' stringa (default: "")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità

|+windows|}
Se diversa dalla stringa nulla, quest'opzione determina il contenuto della riga delle linguette in cima alla finestra Vim. Se non si specifica nulla, Vim userà un default. Vedere |**setting-tabline**| per informazioni ulteriori.

La riga delle linguette compare solo se specificato nell'opzione 'showtabline' e solo quando non è presente una riga linguette gestita dalla GUI. Quando 'e' è presente in 'guioptions' e la GUI supporta una riga delle linguette, si usa invece 'guitablabel'. Notare che le due righe delle linguette sono molto differenti.

Il valore viene valutato come si fa con 'statusline'. È possibile usare |**tabpagenr()**|, |**tabpagewinnr()**| e |**tabpagebuflist()**| per decidere il testo da visualizzare. Usare "%1T" per la prima etichetta, "%2T" per la seconda, etc. Usare elementi "%X" per le etichette di chiusura.

Non dimenticate che solo una delle pagine delle linguette è quella corrente, le altre sono invisibili, e non potete saltare direttamente nelle rispettive finestre.

'tabpagemax' *'tpm'*
'tabpagemax' 'tpm' numero (default: 10)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |+windows|}
Massimo numero di linguette da aprire con l'argomento di Vim |**-p**| o con il comando ":tab all". |**tabpage**|

'tabstop' *'ts'*
'tabstop' 'ts' numero (default: 8)
locale al buffer
Numero di spazi a cui <Tab> equivale nel file. Vedere anche il comando |**:retab**|, e l'opzione 'softtabstop'.

Nota: Impostare 'tabstop' a qualsiasi valore diverso da 8 può far sì che il vostro file sembri sbagliato in altri posti (ad es., quando lo si stampa).

Ci sono quattro modi principali di usare la tabulazione in Vim:
1. tenere sempre 'tabstop' a 8, impostare 'softtabstop' e 'shiftwidth' a 4 (o 3 o quel che preferite) e usare 'noexpandtab'. Così Vim userà una mistura di <TAB> e spazi, ma immettendo <TAB> e <BS> si comporterà come se una <TAB> appaia ogni 4 (o 3) caratteri.
2. Impostare 'tabstop' e 'shiftwidth' ai valori preferiti e usare 'expandtab'. In questo modo inserirete sempre degli spazi. La formattazione non andrà in crisi se poi cambiate 'tabstop'.
3. Impostare 'tabstop' e 'shiftwidth' ai valori preferiti e usare una |**modeline**| per impostare questi stessi valori la prossima volta che andrete a modificare lo stesso file. Può funzionare solo se modificate il file esclusivamente con Vim.
4. Impostare sempre 'tabstop' e 'shiftwidth' allo stesso valore, e 'noexpandtab'. Questo dovrebbe funzionare (solo per la indentatura iniziale) per qualsiasi valore di 'tabstop' venga impostato. Sarebbe bello che le <TAB> inserite dopo il primo carattere diverso dallo spazio fossero però inserite usando degli spazi. Altrimenti dei commenti allineati non lo sarebbero più se si cambia 'tabstop'.

'tagbsearch' *'tbs'* *'notagbsearch'* *'notbs'*
'tagbsearch' 'tbs' booleana (default: on)
globale
{non in Vi}
Quando di cerca una tag (ad es., per il comando |**:ta**|), Vim può usare una ricerca binaria o una ricerca sequenziale in un file di tag. La ricerca binaria trova il tag MOLTO più velocemente, ma una ricerca sequenziale troverà più tag se il file dei tag non è in ordine alfabetico. Vim suppone che i vostri file di tag siano ordinati, o che contengano l'indicazione che non sono ordinati. Solo se non è questo il caso, l'opzione 'tagbsearch' deve essere inattivata.

Quando 'tagbsearch' è attiva, la ricerca binaria viene usata per prima nei file di tag. In alcune situazioni, Vim effettuerà una ricerca sequenziale per certi file, o cercherà nuovamente tutti i file con una ricerca sequenziale. Quando 'tagbsearch' è inattivato, verranno effettuate solo delle ricerche sequenziali.

Una ricerca sequenziale è in ogni caso effettuata, per un file, se a inizio file Vim trova una riga che indica che il file non è ordinato: >

```
!_TAG_FILE_SORTED	0	/qualche commento/
```

< [Lo spazio bianco prima e dopo lo '0' deve essere un solo <Tab>]

Quando è stata effettuata una ricerca binaria senza trovare alcuna corrispondenza in alcuno dei file listati nell'opzione 'tags', e il maiusolo/minuscolo è ignorato, o un'espressione da cercare è utilizzata al posto di un semplice nome di tag, viene effettuato un secondo tentativo con una ricerca sequenziale. Solo in questo secondo caso verranno trovate tag contenute in file non ordinati, e corrispondenze con lettere maiuscole/minuscole diverse da quelle specificate.

Se un file di tag indica che è ordinato ignorando maiuscole e minuscole ['A' e 'a' considerate come una sola lettera - NdT] la ricerca sequenziale si può evitare nel caso si ignorino maiuscole/minuscole. Usate un valore di '2' nella riga "!_TAG_FILE_SORTED" per ottenere questo. Un file di tag può essere ordinato ignorando maiuscole e minuscole usando il flag -f del comando "sort" nella maggioranza dei sistemi Unix, come nel comando "sort -f -o tags tags". Per il programma "Exuberant ctags" versione 5.x o superiore (almeno 5.5) l'opzione --sort=foldcase si può anche usare a questo scopo. Nota Il file viene considerato come se fosse tutto in maiuscolo, altrimenti la cosa non funziona.

Per default, le ricerche di tag sono sensibili al maiuscolo/minuscolo. La differenza maiuscolo/minuscolo è ignorata quando 'ignorecase' è impostato e 'tagcase' vale "followic" o quando 'tagcase' vale "ignore".
Lo stesso accade se 'tagcase' vale "followscs" e 'smartcase' è attivo, oppure se 'tagcase' vale "smart", e l'espressione di ricerca contiene solo caratteri minuscoli.

Quando 'tagbsearch' è a off, la ricerca di una tag è più lenta quando una corrispondenza esiste, ma più veloce se non c'è corrispondenza alcuna. Tag contenute in file non ordinati possono essere trovate solo quando 'tagbsearch' è a off.
Quando il file di tag non è ordinato, o se è ordinato nella maniera sbagliata (non nell'ordine dei byte in codifica ASCII), 'tagbsearch' dovrebbe essere a off, o la riga menzionata più sopra dovrebbe essere inclusa nel file di tag.
Quest'opzione non riguarda comandi che trovano tutti i tag corrispondenti a una ricerca (ad es., il completamento della riga-comandi e il comando ":help").
{Vi: usa sempre la ricerca binaria in alcune versioni}

'tagcase' *'tc'*

'tagcase' 'tc' stringa (default "followic")
globale o locale al buffer |**global-local**|
{non in Vi}

Quest'opzione specifica come gestire il maiuscolo/minuscolo durante la ricerca di un file di tag:

followic	Rispetta l'opzione 'ignorecase'
followscs	Rispetta le opzioni 'smartcase' e 'ignorecase'
ignore	Ignora maiuscolo/minuscolo
match	Cerca corrispondenza esatta
smart	Ignora maiuscolo/minuscolo tranne quando nell'espressione di ricerca compare almeno una lettera maiuscola

NOTA: Quest'opzione è impostata al valore di default di Vi quando si attiva 'compatible' e al valore di default di Vim quando si imposta a off 'compatible'.

'taglength' *'tl'*

'taglength' 'tl' numero (default: 0)

globale
Se diversa da zero, i tag sono significativi solo fino al numero specificato di caratteri.

'tagrelative' *'tr'* *'notagrelative'* *'notr'*
'tagrelative' 'tr' booleana (default Vim: on, default Vi: off)
globale
{non in Vi}
Se l'opzione è a on e si usa un tag file che sta in un'altra directory i nomi di file in quel tag file sono relativi alla directory in cui si trova il tag file stesso.
NOTA: Quest'opzione è impostata al valore di default di Vi impostando 'compatible' e al valore di default di Vim quando 'compatible' viene messa a off.

'tags' *'tag'* *E433*
'tags' 'tag' stringa (default: "./tags,tags", quando compilato con |+emacs_tags|: "./tags,./TAGS,tags,TAGS")
globale o locale al buffer |**global-local**|
Nomi di file da usare per il comando tag, separati da spazi o virgola. Per includere uno spazio o una virgola in un nome di file, premettetegli un backslash ("\").
Vedere |**option-backslash**| per inserire spazi e backslash.
Quando un nome file inizia per "./", il '.' è rimpiazzato con il percorso completo del file corrente. Ma solo quando il flag 'd' non è inclusa in 'cpoptions'. Le variabili d'ambiente vengono valutate |**:set_env**|. Vedere anche |**tags-option**|.
"*", "**" e altre espressioni si possono usare per cercare file di tag in un albero di directory. Vedere |**file-searching**|. Per esempio, "/lib/**/tags" troverà tutti i file di nome "tags" sotto "/lib". Il nome-file vero e proprio non può contenere espressioni regolari, viene usato così come è scritto. Per esempio, "/lib/**/tags?" troverà dei file che si chiamano "tags?". {non disponibile se compilato senza la funzionalità |**+path_extra**|}
La funzione |**tagfiles()**| può essere usata per ottenre una lista dei nomi di fil correntemente in uso.
Se Vim era stato compilato con la funzionalità |**+emacs_tags**|, sono anche supportati file di tag in stile Emacs. Essi sono riconosciuti automaticamente. Il valore di default diviene "./tags,./TAGS,tags,TAGS", a meno che le differenze maiuscolo/minuscolo vengano ignorate (come in MS-Windows).
Vedere |**emacs-tags**|.
L'uso di |**:set+=**| e |**:set-=**| va preferito quando si aggiungono o tolgono nomi di file dalla lista. Ciò permette di evitare problemi laddove una futura versione usasse un altro default.
{Il default di Vi è: "tags /usr/lib/tags"}

'tagstack' *'tgst'* *'notagstack'* *'notgst'*
'tagstack' 'tgst' booleana (default: on)
globale
{non in tutte le versioni di Vi}
Se impostata, la pila dei tag (|**tagstack**|) è usata normalmente. Se a off, un comando ":tag" o ":tselect" con un argomento non invia la tag alla pila dei tag. Un successivo comando ":tag" senza un argomento, un comando ":pop" o qualsiasi altro comando che usi la pila dei tag userà la pila dei tag non modificata, ma cambierà il puntatore all'elemento attivo.
Annullare quest'opzione è utile se si usa un comando ":tag" in una mappatura che dovrebbe lasciare inalterata la pila dei tag.

'tcldll'
'tcldll' stringa (default: dipende dalla compilazione di Vim)
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |**+tcl/dyn**|}
Specifica il nome della libreria condivisa Tcl. Il default è DYNAMIC_TCL_DLL, specificato al momento della compilazione.
Le variabili d'ambiente sono valutate |**:set_env**|.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'term' *E529* *E530* *E531*

'term' stringa (Il default è il valore di $TERM. Se assente:
nella GUI: "builtin_gui"
in Amiga: "amiga"
in BeOS: "beos-ansi"
in Mac: "mac-ansi"
in MiNT: "vt52"
in MS-DOS: "pcterm"
in OS/2: "os2ansi"
in Unix: "ansi"
in VMS: "ansi"
in Win 32: "win32")
globale
Nome del terminale. Usato per scegliere i caratteri di controllo del terminale. Le variabili d'ambiente vengono valutate |:set_env|.
Ad esempio: >
:set term=$TERM
< Vedere |termcap|.

'termbidi' *'tbidi'*
'notermbidi' *'notbidi'*
'termbidi' 'tbidi' booleana (default: off, on per "mlterm")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità |+arabic|}
Il terminale si occupa della bi-direzionalità del testo (come specificata da Unicode). È anche a carico del terminale assumere la forma richiesta da alcuni lingue (come l'Arabo).
Impostare quest'opzione implica che 'rightleft' non vada impostato quando si imposta 'arabic' e che il valore di 'arabicshape' verrà ignorato.
Nota Impostare 'termbidi' ha come effetto immediato quello di far ignorare 'arabicshape', ma 'rightleft' non è cambiato automaticamente.
Quest'opzione è messa a off quando la GUI parte.
Per dettagli ulteriori vedere |arabic.txt|.

'termencoding' *'tenc'*
'termencoding' 'tenc' stringa (default: ""; con la GUI GTK+: "utf-8"; con la GUI Macintosh: "macroman")
globale
{disponibile solo se compilato con la funzionalità |+multi_byte|}
{non in Vi}
Codifica usata per il terminale. Questo specifica che codifica caratteri la tastiera produce e il video riesce a visualizzare. Per la GUI, si applica solo alla tastiera ('encoding' è usato per il video). Questo non vale per il Mac: quando 'macatsui' è a off, allora 'termencoding' dovrebbe essere "macroman". *E617*
Nota: Questo non vale per la GUI GTK+. Dopo che la GUI è stata inizializzata con successo, 'termencoding' è messo obbligatoriamente a "utf-8". Ogni tentativo di impostarlo a un valore diverso sarà rifiutato, e verrà emesso un messaggio di errore.
Per le versioni Console e GUI Win32 'termencoding' non è usato, perché il sistema Win32 fornisce sempre caratteri Unicode.
Se nullo, si usa la stessa codifica specificata nell'opzione 'encoding'.
Questo è il valore normale.
Non tutte le combinazioni di 'termencoding' e 'encoding' sono valide. Vedere |encoding-table|.
Il valore di quest'opzione deve essere supportato dalla conversione interna o da iconv(). Quando non è possibile farlo, non sarà effettuata alcuna conversione, e probabilmente si andrà incontro a dei problemi con i caratteri non-ASCII.
Esempio: State lavorando con il "locale" impostato a euc-jp (Giapponese) e volete editare un file codificato in UTF-8: >
:let &termencoding = &encoding
:set encoding=utf-8
< È necessario fare questo se il vostro sistema operativo noon ha supporto "locale" per UTF-8.

'termguicolors' *'tgc'* *E954*
'termguicolors' 'tgc' booleana (default: off)

globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+termguicolors**|}
Se impostata, usa gli attributi |**highlight-guifg**| e |**highlight-guibg**|
per il terminale (utilizzando quindi colori a 24 bit).

Funziona solo se il terminale è compatibile con lo standard
ISO-8613-3. Se impostare questa funzione non funziona (se viene
visualizzata un'interfaccia utente non colorata) leggere
|**xterm-true-color**| può essere di aiuto per risolvere il problema.

Per la console Win32, è necessaria una versione di Windows 10 1703
(Creators Update) o successiva. Usare questo controllo per
determinare se la funzionalità è disponibile: >
```
if has('vcon')
```
< Perché questo funzioni, Vim dev'essere generato con la funzionalità
|**+vtp**|.

Si noti che sono ancora in uso gli attributi di "cterm", non
quelli che si riferiscono alla "gui".
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'termwinscroll' *'twsl'*
'termwinscroll' 'twsl' numero (default: 10000)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+terminal**|}
Numero di linee di scorrimento all'indietro da ricordare. Quando si
supera questo limite, il primo 10% delle linee di scorrimento viene
cancellato. Lo scopo è quello di ridurre l'utilizzo di memoria.
Vedere |**Terminal-Normal**|.

'termwinkey' *'tk'*
'termwinkey' 'twk' stringa (default: "")
locale alla finestra
{non in Vi}
Il carattere che inizia un comando CTRL-W in una finestra a
terminale. Gli altri caratteri sono passati al lavoro in
esecuzione nella finestra.
La stringa dev'essere composta da un solo carattere, ma può
essere composta da più di un byte.
Si può usare la notazione <>, p.es.: >
```
:set termwinkey=<C-L>
```
< La stringa dev'essere un singolo tasto, ma può consistere in più
di un byte.
Se l'opzione non è impostata, si usa CTRL-W, in modo che CTRL-W :
consente di accedere alla riga dei comandi.
Se 'termwinkey' vale CTRL-L, sarà immettendo CTRL-L : che si accede
alla riga dei comandi.

'termsize' *'tms'*
'termwinsize' 'tws' stringa (default: "")
locale alla finestra
{non in Vi}
Dimensione della finestra di terminale (|**terminal**|).
Formato: {righe}x{colonne} o {righe}*{colonne}.
- Se non si specifica nulla, il terminale ottiene le dimensioni dalla
finestra.
- Se impostata usando il segno "x" (p.es., "24x80") la dimensione del
terminale non viene adattata a quella della finestra. Se la
finestra ha dimensioni inferiori, solo la parte superiore sinistra
viene visualizzata.
- Se impostata usando il segno "*" (p.es., "10x50") la dimensione del
terminale viene adattata a quella della finestra, ma non sarà più
piccola del numero specificato di righe e/o colonne.
- Quando il numero di righe è zero, si usa l'altezza della finestra.
- Quando il numero di colonne è zero, si usa la larghezza della
finestra.
- Specificare "0x0" o "0*0" è lo stesso che non specificare nulla.

Esempi:
"30x0" usa 30 righe e la larghezza corrente della finestra.
"20*0" usa almeno 20 righe e la larghezza corrente della finestra.
"0*40" usa l'altezza corrente della finestra e almeno 40 colonne.
NOTA Il comando eseguito nella finestra di terminale può ancora modificare la dimensione del terminale stesso. In tal caso la finestra di Vim verrà adeguata alla dimensione richiesta, se possibile.

'terse' *'noterse'*
'terse' booleana (default: off)
globale
Quando impostato: aggiunge il flag 's' all'opzione 'shortmess' (questo fa sì che il messaggio per una ricerca che oltrepassi l'inizio o la fine del file non venga visualizzato). Quando impostato a off: toglie il flag 's' dall'opzione 'shortmess'. {Vi abbrevia molti messaggi}

'textauto' *'ta'* *'notextauto'* *'nota'*
'textauto' 'ta' booleana (default Vim: on, default Vi: off)
globale
{non in Vi}
Quest'opzione è obsoleta. Usare 'fileformats'.
Per compatibilità all'indietro, quando 'textauto' è impostato, 'fileformats' è impostato al valore di default del sistema corrente.
Quando 'textauto' è messo a off, 'fileformats' è reso nullo.
NOTA: Quest'opzione è impostata al valore di default di Vi impostando 'compatible' e al valore di default di Vim quando 'compatible' viene messa a off.

'textmode' *'tx'* *'notextmode'* *'notx'*
'textmode' 'tx' booleana (MS-DOS, Win32 e OS/2: on per default, altri: off per default)
locale al buffer
{non in Vi}
Quest'opzione è obsoleta. Usare 'fileformat'.
Per compatibilità all'indietro, quando 'textmode' è impostato, 'fileformat' è impostato a "dos". Quando 'textauto' è messo a off, 'fileformat' è impostato a "unix".

'textwidth' *'tw'*
'textwidth' 'tw' numero (default: 0)
locale al buffer
{non in Vi}
Massima larghezza del testo che si sta inserendo. Una riga più lunga verrà spezzata dopo uno spazio bianco, per rispettare la larghezza impostata. Un valore di zero disabilita questa modo di procedere.
'textwidth' è messa a zero quando si imposta l'opzione 'paste' e ripristinata quando 'paste' è messa a off.
Quando 'textwidth' è zero, si può usare 'wrapmargin'. Vedere anche 'formatoptions' e |**ins-textwidth**|.
Quando 'formatexpr' è attiva, sarà usata per spezzare la riga.
NOTA: Quest'opzione è impostata a 0 se si attiva 'compatible'.

'thesaurus' *'tsr'*
'thesaurus' 'tsr' stringa (default: "")
globale o locale al buffer |**global-local**|
{non in Vi}
Lista di nomi di file, separati da virgola, usati per cercare parole di "thesaurus" da usare nei comandi di completamento parola |**i_CTRL-X_CTRL-T**|. Ogni riga di file dovrebbe contenere parole con significato simile, separate da caratteri che non siano contenuti nelle parole (lo spazio bianco è consigliato). La massima lunghezza di una riga è 510 byte.
Per ottenere un file da usare per questo, controllare le FAQ per le "wordlist" (liste di parole) al sito:
[Purtroppo il link qui sotto non è più disponibile, quale potrebbe essere un sito alternativo?]
ftp://ftp.ox.ac.uk/pub/wordlists/ Scaricare innanzitutto il file README.
Per includere una virgola in un nome di file, premettetegli un backslash ("\"). Gli spazi dopo una virgola sono ignorati, ma in altre posizioni, sono considerate parte del nome del file.

Vedere |**option-backslash**| per inserire spazi e backslash.
L'uso di |**:set+=**| e |**:set-=**| va preferito quando si aggiungono o tolgono nomi di directory dalla lista. Ciò permette di evitare problemi laddove una futura versione usasse un altro default. Comandi contenuti fra "`" (backtick) non sono permessi in questa opzione per motivi di sicurezza.

'tildeop' *'top'* *'notildeop'* *'notop'*
'tildeop' 'top' booleana (default: off)
globale
{non in Vi}
Se impostata: Il comando tilde "~" si comporta come un operatore.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'timeout' *'to'* *'notimeout'* *'noto'*
'timeout' 'to' booleana (default: on)
globale
'ttimeout' *'nottimeout'*
'ttimeout' booleana (default: off, impostata a on in
|**defaults.vim**|)
globale
{non in Vi}
Queste due opzioni determinano congiuntamente il comportamento quando una parte di una sequenza di tasti mappata è stata immessa:

'timeout'	'ttimeout'	azione ~
off	off	non ignorare dopo qualche tempo (time-out)
on	on o off	ignora dopo qualche tempo (time-out) :mappings e key-code ["key-code" è un tasto immesso tramite descrizione lunga, ad es., scrivendo <F12> per indicare il tasto marcato F12 - NdT]
off	on	ignora dopo qualche tempo (time-out) key-code

Se entrambe le opzioni sono impostate a off, Vim attende finché la sequenza completa di mappature o il key-code sia stato immesso, oppure fin quando è chiaro che non esiste una mappatura o un tasto corrispondente ai caratteri immessi. Per esempio, se si è mappato "vl" e Vim ha ricevuto 'v', il carattere successivo è necessario per capire se òa 'v' è seguita da una 'l'.
Se entrambe le opzioni sono impostate a on, Vim attende per circa un secondo l'immissione del carattere successivo. Dopo quel tempo, i caratteri già ricevuti sono interpretati come caratteri singoli. Il tempo di attesa può essere modificato con l'opzione 'timeoutlen'.
Su terminali lenti o computer utilizzati intensamente un time-out può causare problemi di malfunzionamento dei tasti che muovono il cursore.
Se entrambe le opzioni sono a off, Vim aspetta per l'eternità dopo che si è battuto il stato <Esc>, se ci sono dei key-code [maniere alternative di immettere un carattere] che iniziano con <Esc>.
Per uscirne, battere <Esc> una seconda volta. Se non avete problemi con qualche key-code, ma volete che delle sequenze di tasti mappate (:mapped) non vadano in time-out dopo 1 secondo, impostate a on l'opzione 'ttimeout' e impostate a off l'opzione 'timeout'.

NOTA: 'ttimeout' è impostata a off se si attiva 'compatible'.

'timeoutlen' *'tm'*
'timeoutlen' 'tm' numero (default: 1000)
globale
{non in tutte le versioni di Vi}
'ttimeoutlen' *'ttm'*
'ttimeoutlen' 'ttm' numero (default: -1, impostata a 100 in
|**defaults.vim**|)
globale
{non in Vi}
Tempo di attesa in millisecondi perché sia completata una sequenza di caratteri che immette un key-code, oppure una sequenza di tasti mappata. Si usa anche per CTRL-\ CTRL-N e CTRL-\ CTRL-G, quando una parte di un comando sia stata immessa.
Normalmente solo 'timeoutlen' è in uso, e 'ttimeoutlen' vale -1.
Quando si desidera un valore di time-out differente per i key-code,

impostare 'ttimeoutlen' a un numero non negativo.

ttimeoutlen	ritardo mappatura	ritardo key-code
< 0	'timeoutlen'	'timeoutlen'
>= 0	'timeoutlen'	'ttimeoutlen'

Il time-out interviene solo quando lo stabiliscono le opzioni 'timeout' e 'ttimeout'. Un'impostazione utile potrebbe essere >

```
:set timeout timeoutlen=3000 ttimeoutlen=100
```

< (time-out sulle mappature dopo 3 secondi, time-out sui key-code dopo un decimo di secondo).

'title' *'notitle'*

'title' booleana (inattiva per default, a on se il titolo può essere ripristinato)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+title**|}

Se impostata, il titolo della finestra sarà impostata al valore di 'titlestring' (se specificata), oppure a:

```
nome_file [+=-] (percorso) - VIM
```

Dove:

nome_file	il nome del file che stiamo modificando
-	indica che il file non può essere modificato, 'ma' è a off
+	indica che il file è stato modificato
=	indica che il file è in sola lettura
=+	indica che il file è in sola lettura, ma è stato modificato
(percorso)	è il percorso che conduce al file in modifica [nome della directory]
- VIM	Il nome del server |**v:servername**| oppure "VIM"

Funziona solo se il terminale supporta l'impostazione di titoli per la finestra (al momento, Amiga console, Win32 console, tutte le versioni GUI e i terminali con l'opzione 't_ts' non nulla, ossia i terminali Unix e iris-ansi per default, dove 't_ts' è presa dalle "termcap" include nel sistema).

X11

Quando Vim è stato compilato con la definizione di HAVE_X11, il titolo originale verrà ripristinato, se possibile. L'output del comando ":version" includerà "+X11" quando HAVE_X11 è stato definito, altrimenti verrà visualizzato "-X11". Questo vale anche per il nome dell'icona |'icon'|.
Ma: quando Vim è eseguito con l'argomento |**-X**|, il ripristino del titolo non funzionerà (tranne che Vim venga eseguito nella GUI).
Se il titolo non può essere ripristinato, è impostato al valore di 'titleold'. In questo caso potreste voler ripristinare il titolo al di fuori di Vim.
Quando si usa un xterm collegandosi con una macchina remota potete usare questo comando:

```
    rsh nome_della_macchina_remota xterm -display $DISPLAY &
```

a quel punto la variabile d'ambiente WINDOWID dovrebbe essere resa disponibile il titolo della finestra dovrebbe essere cambiato a quel che doveva essere, dopo che siete usciti da Vim.

'titlelen'

'titlelen' numero (default: 85)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+title**|}

Specifica la percentuale di 'columns' (colonne) da usare per la lunghezza del titolo della finestra. Quando il titolo è più lungo, solo la parte finale del nome del file è visualizzata. Un carattere '<' prima del nome del file è usato per segnalare questo.
L'uso di una percentuale fa sì che la lunghezza del titolo si adatti all'ampiezza della finestra. Ma non funzionerà perfettamente, perché il numero effettivo di caratteri disponibile dipende anche dal font utilizzato e da altri elementi presenti nella barra del titolo.
Quando 'titlelen' è a zero si usa il nome intero del file. Altrimenti si possono specificare valori da 1 a 30000 percento.
'titlelen' è anche usato per l'opzione 'titlestring'.

'titleold'
'titleold' stringa (default: "Grazie per aver volato con Vim")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+title|}
Quest'opzione sarà usata come titolo della finestra quando si esce da Vim, se il titolo originale non può essere ripristinato. Valido solo se 'title' è attivato o 'titlestring' non è una stringa nulla. Quest'opzione non può essere impostata da |modeline|, o nel |sandbox|, per motivi di sicurezza.

'titlestring'
'titlestring' stringa (default: "")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|+title|}
Quando quest'opzione non è nulla, sarà usata per il titolo della finestra. Questo vale solo quando l'opzione 'title' è a on. Funziona solo se il terminale supporta l'impostazione di titoli per la finestra (al momento, Amiga console, Win32 console, tutte le versioni GUI e i terminali con l'opzione 't_ts' non nulla). Quando Vim è stato compilato con la definizione di HAVE_X11, il titolo originale verrà ripristinato se possibile, vedere |X11|. Quando quest'opzione contiene elementi in stile "printf" ('%'), questi saranno valutati con le stesse regole usate per 'statusline'. Esempio: >
:auto BufEnter * let &titlestring = hostname() . "/" . expand("%:p")
:set title titlestring=%<%F%=%l/%L-%P titlelen=70
< il valore di 'titlelen' è usato per allineare gli elementi non mezzo o alla destra dello spazio disponibile. C'è chi preferisce avere il nome del file come prima cosa: >
:set titlestring=%t%(\ %M%)%(\ (%{expand(\"%:~:.:h\")})%)%(\ %a%)
< Nota: Si noti l'uso of "%{ }" e di un'espressione per ottenere il nome del file, senza usare il nome del file stesso. La stringa "%(%)" permette di aggiungere spazi di separazione solo se necessari. NOTA: L'uso di caratteri speciali in 'titlestring' può far sì che la visualizzazione sia incorretta (ad es., quando contiene un carattere <CR> o <NL>).
{non disponibile se compilato senza la funzionalità |+status_line|}

'toolbar' *'tb'*
'toolbar' 'tb' stringa (default: "icons,tooltips")
globale
{solo per |+GUI_GTK|, |+GUI_Athena|, |+GUI_Motif| e
|+GUI_Photon|}
I valori di quest'opzione controllano varie impostazioni della toolbar. I valori possibili sono:

icons	I bottoni della toolbar sono visualizzati come icone.
text	I bottoni della toolbar sono visualizzati
horiz	Icone e testi in un bottone della toolbar sono disposti orizzontalmente. {solo nella GUI GTK+ 2}
tooltips	Dei suggerimenti vendono visualizzati per ogni bottone della toolbar.

Il termine "tooltips" indica il testo che viene visualizzato dopo aver posto il cursore del mouse per qualche istante su un bottone della toolbar.

Per far sì che la toolbar visualizzi sia icone che testi, usare: >
:set tb=icons,text
< Motif e Athena non sono in grado di visualizzare icone e testi allo stesso tempo. Richiedendoli entrambi, sono le icone saranno visualizzate.

Se nessuna delle stringhe specificate in 'toolbar' è valida, o se 'toolbar' non è impostato, quest'opzione è ignorata. Per non visualizzare del tutto la toolbar, occorre usare l'opzione 'guioptions'. Ad esempio: >
:set guioptions-=T

< Vedere anche |**gui-toolbar**|.

'toolbariconsize' *'tbis'*
'toolbariconsize' 'tbis' stringa (default: "small")
globale
{non in Vi}
{solo nella GUI GTK+}
Controlla la grandezza delle icone della toolbar. I valori possibili sono:

tiny	Usare icone molto piccole.
small	Usare icone piccole (default).
medium	Usare icone medie.
large	Usare icone grandi.
huge	Usare icone ancora più grandi.
giant	Usare icone molto grandi.

Le dimensioni esatte in pixel delle varie grandezze delle icone dipendono dallo stile di visualizzazione ("theme") corrente. Dimensioni abbastanza frequenti sono giant=48x48, huge=32x32, large=24x24, medium=24x24, small=20x20 e tiny=16x16.

Se 'toolbariconsize' non è impostato, viene usata la dimensione globale di default come definita dalle preferenze dell'utilizzatore o dallo stile di visualizzazione.

'ttybuiltin' *'tbi'* *'nottybuiltin'* *'notbi'*
'ttybuiltin' 'tbi' booleana (default: on)
globale
{non in Vi}
Se impostata, le "termcaps" interne a Vim sono usate prima di cercarne di esterne.
Se inattiva, le "termcaps" interne a Vim sono usate dopo aver cercato quelle esterne.
Quando quest'opzione è cambiata, dovreste subito dopo impostare l'opzione 'term' per rendere effettiva la modifica, ad es.: >

```
:set notbi term=$TERM
```

< Vedere anche |**termcap**|.
Motivazione: Il default per quest'opzione è "on", perché le "termcaps" interne a Vim sono in generale migliori (molti sistemi contengono descrizioni di xterm errate...).

'ttyfast' *'tf'* *'nottyfast'* *'notf'*
'ttyfast' 'tf' booleana (inattiva per default, a on quando 'term' è xterm, hpterm, sun-cmd, screen, rxvt, dtterm o iris-ansi; pure a on quando si esegue Vim in una console DOS)
globale
{non in Vi}
Indica una connessione di terminale veloce. Un numero maggiore di caratteri verrà inviato allo schermo in caso di rinfresco dell'immagine, invece di usare i comandi per inserire/cancellare righe singole. Migliora la qualità del rinfresco immagine quando sono aperte più finestre e il terminale non supporta una regione per lo scorrimento.
Inoltre viene abilitata la scrittura di caratteri in più alla fine di ogni riga sullo schermo per righe che si estendono oltre la lunghezza della riga della schermata. Questo viene utile quando si usa copia/incolla per mezzo del mouse in un "xterm" e in altri tipi di terminale.

'ttymouse' *'ttym'*
'ttymouse' 'ttym' stringa (Il default dipende da 'term')
globale
{non in Vi}
{solo in Unix e VMS, non funziona nella GUI; non disponibile se compilato senza la funzionalità |**+mouse**|}
Nome del tipo di terminale per il quale vanno riconosciuti i codici del mouse.
Per ora, si possono specificare solo le seguenti stringhe:
xterm-mouse

xterm	gestione mouse in maniera simile a "xterm". Il mouse genera "<Esc>[Mscr", dove "scr" è composto da tre byte:

"s" = status del bottone
"c" = colonna più 33
"r" = riga più 33
Questo funziona solo fino a un massimo di 223 colonne!
Vedere "dec", "urxvt" e "sgr" per soluzioni.

xterm2 Come "xterm", ma con "xterm" che rende nota la posizione del mouse mentre il mouse è in movimento. Questo consente un funzionamento molto più veloce e preciso. Il vostro programma "xterm" deve essere almeno al livello di patch 88 / XFree 3.3.3 perché possa funzionare in questo modo. Vedere sotto come Vim riesce a determinare questo caso automaticamente.
netterm-mouse

netterm Gestione mouse di tipo "NetTerm". Il mouse genera "<Esc>}r,c<CR>", dove "r,c" sono due numeri decimali che indicano la riga e la colonna.
dec-mouse

dec Gestione mouse di tipo DEC. Il mouse genera una sequenza di caratteri piuttosto complessa, che inizia con "<Esc>[".
Questa gestione è possibile anche per un Xterm, se era stato configurato con "--enable-dec-locator".
jsbterm-mouse

jsbterm Gestione mouse di tipo JSB.
pterm-mouse

pterm Gestione mouse di tipo QNX.
urxvt-mouse

urxvt Gestione mouse per il terminale urxvt (rxvt-unicode). Il mouse funziona solo se il terminale supporta questo stile di codifica, ma non è limitato a 223 colonne, a differenza di "xterm" o "xterm2".
sgr-mouse

sgr Gestione mouse per il terminale che rilascia informazioni in stile SGR (Select Graphic Rendition). Il mouse funziona anche in colonne oltre la 223. Quest'opzione è compatibile all'indietro con "xterm2" perché è in grado di decodificare codice del mouse in stile "xterm2".

La gestione del mouse deve essere abilitata al momento della compilazione di Vim |**+mouse_dec**| |**+mouse_netterm**| |**+mouse_jsbterm**| |**+mouse_urxvt**| |**+mouse_sgr**|.
Solo "xterm2" è veramente individuato al momento. I codici del mouse NetTerm sono sempre riconosciuti, se Vim è stato abilitato in fase di compilazione. I codici del mouse DEC sono riconosciuti se Vim è stato abilitato in fase di compilazione, e 'ttymouse' non è "xterm", "xterm2", "urxvt" o "sgr" (perché i codici mouse DEC sono in conflitto con essi).
Quest'opzione è automaticamente impostata a "xterm", quando l'opzione 'term' è impostata a un nome che inizia per "xterm", "mlterm", "screen", "tmux", "st" (solo una corrispondenza completa), "st-" o "stterm", e 'ttymouse' non sia già stato impostato.
Inoltre, se Vim è compilato con la funzionalità |**+termresponse**| e |**t_RV**| è impostato alla sequenza di protezione che chiede il numero di versione di xterm, viene utilizzato un codice di determinazione più sofisticato.
Il valore "xterm2" sarà impostato se la versione di xterm risulta essere fra 95 e 276. Il valore "sgr" verrà impostato se la versione di xterm è 277 o superiore e quando Vim determina di essere in ambiente Mac Terminal.app o Iterm2.
Se non si vuole impostare automaticamente 'ttymouse' a "xterm2" o a "sgr", impostare t_RV alla stringa nulla: >

```
:set t_RV=
```

<

'ttyscroll' *'tsl'*

'ttyscroll' 'tsl' numero (default: 999)
globale

Massimo numero di righe per le quali far scorrere lo schermo. Se ci sono da far scorrere più righe, la finestra è rinfrescata completamente. Per terminali in cui lo scorrimento è molto lento e il rinfrescamento completo non è lento, quest'opzione può essere impostata a un numero basso, ad. es. 3, per velocizzare la visualizzazione.

'ttytype' *'tty'*
'ttytype' 'tty' stringa (default: valore di $TERM)
globale
Altro nome per 'term', vedere più sopra.

'undodir' *'udir'*
'undodir' 'udir' stringa (default: ".")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+persistent_undo|}
Lista di nomi di directory names per file di undo, separate da
virgole.
Vedere |'backupdir'| per dettagli sul formato della lista.
"." vuol dire usare la directory in cui si trova il file. Il nome del
file di undo per "file.txt" è ".file.txt.un~".
Per altre directory, il nome file è il nome completo del file in
modifica, con i separatori ["/" o "\"] rimpiazzati da "%".
In scrittura: La prima directory esistente è usata. "." è sicuramente
utilizzabile, per cui nessuna directory elencata dopo "." sarà usata
in scrittura.
In lettura, tutte le directory in elenco sono controllare, alla
ricerca di un file di undo. Il primo file di undo trovato viene
utilizzato. Se risulta non leggibile, viene emesso un messaggio di
errore, ma non si usa alcun altro file alternativo.
vedere |**undo-persistence**|.

'undofile' *'noundofile'* *'udf'* *'noudf'*
'undofile' 'udf' booleana (default: off)
locale al buffer
{non in Vi}
{disponibile solo se compilato con la funzionalità
|+persistent_undo|}
Quando l'ozione è impostate, Vim automaticamente salva la storia degli
undo in un file di undo, quando scrive il buffer su un file, e può
ripristinare la storia degli undo dallo stesso file se il buffer viene
nuovamente letto.
La directory in cui immagazzinare il file di undo è specificata da
'undodir'.
Per maggiori informazioni riguardo a questa funzionalità, vedere
|**undo-persistence**|.
Il file di undo non è letto quando 'undoreload' chiede che il buffer
prima di un nuovo caricamento del file sia salvato in vista di un
possibile undo.
Quando 'undofile' vale off il file di undo NON è cancellato.
NOTA: Quest'opzione è impostata a off se si attiva 'compatible'.

'undolevels' *'ul'*
'undolevels' 'ul' numero (default: 100, 1000 per Unix, VMS,
Win32 e OS/2)
globale
{non in Vi}
Massimo numero di modifiche che possono essere annullate. Poiché le
informazioni per annullare sono mantenute in memoria, numeri più
elevati richiederanno l'utilizzo di una maggiore quantità di memoria
(e, comunque, può bastare una sola modifica a richiedere una quantità
spropositata di memoria).
Impostare a 0 per mantenere la compatibilità con Vi: un livello di
annullamento, e un secondo "u" annulla se stesso: >
set ul=0
< La compatibilità con Vi può anche essere ottenuta includendo il flag
'u' in 'cpoptions', e in questo caso resta possibile usare CTRL-R per
annullare l'annullamento.
Impostare a un numero negativo per non avere alcuna possibilità di
annullamento. Volendo, si può richiedere questo anche solo
limitatamente al buffer corrente: >
setlocal ul=-1
< Questo può servire nel caso in cui una singola modifica basti a
esaurire la memoria disponibile.

Il valore locale è impostato a -123456 quando si chiede di usare il
valore globale.

Vedere anche |**clear-undo**|.

'undoreload' *'ur'*
'undoreload' 'ur' numero (default: 10000)
globale
{non in Vi}
Salva l'intero buffer in vista di un possibile undo, quando lo si ricarica (quando lo si legge di nuovo da disco). Ciò vale per il comando ":e!" e per un nuovo caricamento nel caso il buffer sia stato cambiato indipendentemente da Vim (da un altro programma).
Vedere |**FileChangedShell**|.
Il salvataggio avviene solo quando quest'opzione ha valori negativi o quando il numero di linee è minore del valore di quest'opzione.
Impostate a zero quest'opzione per disabilitare la possibilità di un undo a fronte di un ricaricamento del file.

Quìando si salva una copia di undo in vista di un ricaricamento, gli eventuali file di undo esistenti non sono letti.

Notare che questo fa sì che l'intero buffer sia manenuto in memoria. Impostate quest'opzione a un valore basso se provoca il consumo di tutta la memoria disponibile.

'updatecount' *'uc'*
'updatecount' 'uc' numero (default: 200)
globale
{non in Vi}
Dopo aver immesso il numero di caratteri specificato da quest'opzione, il file di swap sarà riscritto su disco. Quando quest'opzione vale 0, non viene creato alcun file di swap (vedere il capitolo sul ripristino |**crash-recovery**|). 'updatecount' è impostato a zero se si esegue Vim specificando l'opzione "-n" option, vedere |**startup**|. Quando si edita in modalità di sola lettura, quest'opzione sarà inizializzata a 10000.
Il file di swap può essere disabilitato a livello di singoli buffer, tramite l'opzione|'**swapfile**'|.
Quando 'updatecount' passa da un valore zero a uno maggiore di zero, dei file di swap sono creati per tutti i buffer in cui è impostata l'opzione 'swapfile'. Quando 'updatecount' è impostato a zero, i file di swap esistenti vengono cancellati.
Vedere anche |'**swapsync**'|.
Quest'opzione non ha senso in buffer in cui |'**buftype**'| è "nofile" o "nowrite".

'updatetime' *'ut'*
'updatetime' 'ut' numero (default: 4000)
globale
{non in Vi}
Se il numero di millisecondi specificato è trascorso senza che sia stato immesso alcun carattere, il file di swap sarà riscritto su disco (vedere |**crash-recovery**|). Usato anche dall'autocomando dell'evento |**CursorHold**|.

'verbose' *'vbs'*
'verbose' 'vbs' numero (default: 0)
globale
{non in Vi, sebbene alcune versioni abbiano un'opzione booleana 'verbose'}
Se impostata a un valore maggiore di 0, Vim emette dei messaggi riguardo a quel che sta facendo.
Al momento, questi sono i messaggi emessi:
>= 1 Quando il file "viminfo" è letto o scritto.
>= 2 Quando un file viene eseguito tramite ":source".
>= 5 Per ogni file di tag e di include esaminato.
>= 8 File per i quali viene eseguito un gruppo di autocomandi.
>= 9 Ogni autocomando eseguito.
>= 12 Ogni funzione eseguita.
>= 13 Quando un'interruzione per eccezione è rifiutata, accettata, terminata o scartata.
>= 14 Se c'è qualcosa in sospeso in una clausola ":finally".
>= 15 Ogni comando Ex eseguito (troncato dopo 200 caratteri).

Quest'opzione si può impostare anche con l'argomento "-V".

Vedere |**-V**|.
Quest'opzione è anche impostata dal comando |**:verbose**|.

When the 'verbosefile' option is set then the verbose messages are not displayed.

'verbosefile' *'vfile'*
'verbosefile' 'vfile' stringa (default: "")
globale
{non in Vi}
Quando l'opzione non è nulla, tutti i messaggi sono scritti su un file dal nome che si specifica.
Se il file esiste, i messaggi sono aggiunti in fondo.
La scrittura del file termina quando si esce da Vim o quando l'opzione 'verbosefile' ridiventa la stringa nulla. Le scritture a disco avvengono solo quando un buffer è pieno, quindi possono non essere visibili immediatamente.
Impostare 'verbosefile' a un valore nuovo equivale a svuotarlo prima.
La differenza con |**:redir**| è che i messaggi verbosi non sono visualizzati (al terminale) quando 'verbosefile' è impostato.

'viewdir' *'vdir'*
'viewdir' 'vdir' stringa (default per Amiga, MS-DOS, OS/2 e Win32:
"$VIM/vimfiles/view",
per Unix: "~/.vim/view",
per Macintosh: "$VIM:vimfiles:view"
per VMS: "sys$login:vimfiles/view"
per RiscOS: "Choices:vimfiles/view")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+mksession**|}
Nome della directory in cui depositare file per |**:mkview**|.
Quest'opzione non può essere impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.

'viewoptions' *'vop'*
'viewoptions' 'vop' stringa (default: "folds,options,cursor,curdir")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+mksession**|}
Modifica l'effetto del comando |**:mkview**|. È una lista di parole, separate da virgole. Ogni parola consente il salvataggio e il ripristino di qualcosa:

parola	salvataggio e ripristino ~
cursor	posizione del cursore nel file e nella finestra
folds	piegature create manualmente, piegature aperte/chiuse, opzioni locali di piegatura
options	opzioni e mappature locali per una finestra o un buffer (non valori globali per opzioni locali)
localoptions	equivalente a "options"
slash	backslash ("\") in nomi di file, rimpiazzati con la barra ("/") normale
unix	mantenere il formato di fine-riga Unix (un singolo carattere <NL>), anche se si è in Windows o DOS
curdir	la directory locale a livello di finestra, se impostata con il comando `:lcd`

"slash" e "unix" sono utili in Windows quando si condividono con Unix file che si vogliono visualizzare. La versione Unix di Vim non può eseguire col comando ":source" degli scritti di Vim in formato DOS, mentre la versione Windows di Vim può farlo con script in formato Unix.

'viminfo' *'vi'* *E526* *E527* *E528*
'viminfo' 'vi' stringa (default Vi: "", default Vim per MS-DOS,
Windows e OS/2: '100,<50,s10,h,rA:,rB:,
per Amiga: '100,<50,s10,h,rdf0:,rdf1:,rdf2:
per altri: '100,<50,s10,h)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità

|**+viminfo**|}
Quando l'opzione è specificata e diversa dalla stringa nulla, il file viminfo è letto alla partenza di Vim e scritto alla fine della sessione di Vim (vedere |**viminfo-file**|). Ciò non vale quando si specifica come 'viminfofile' NONE.
Il valore dell'opzione dovrebbe essere una lista di parametri, separati da virgola, ognuno dei quali consiste di un singolo carattere, che identifica un particolare parametro, seguito da un numero o da un stringa di caratteri che specificano il valore di tale parametro. Se un dato carattere non è specificato, per il parametro corrispondente si usa il valore di default. Quella che segue è una lista dei caratteri identificativi, e degli effetti che si ottengono specificandone il valore.
CAR. VALORE ~
viminfo-!
! Quando inserito, salva e ripristina le variabili globali che iniziano con lettera maiuscola e non ne contengono una minuscola. Così "KEEPTHIS e "K_L_M" vengono memorizzate, ma "KeepThis" e "_K_L_M" non lo sono. Elementi nidificati di Liste e Dizionari potrebbero non essere ritrovati in modo corretto, e al loro posto essere restituiti elementi vuoti.
viminfo-quote
" Numero massimo di righe salvate per ciascun registro. Vecchio nome dell'elemento '<', con lo svantaggio che occorre mettere una backslash dopo il ", altrimenti esso verrà riconosciuto come l'inizio di un commento!
viminfo-%
% Se inserito salva e ripristina l'elenco dei buffer. Se Vim viene avviato con un nome di file come argomento, l'elenco dei buffer non verrà ripristinato. Se Vim è invocato senza un nome di file, l'elenco dei buffer verrà ripristinato dal file viminfo. Buffer di tipo quickfix ('buftype'), non listati ('buflisted'), senza nome e buffer su supporti rimovibili (|**viminfo-r**|) non sono salvati.
Se seguito da un numero, il numero specifica il numero massimo di buffer da memorizzare. Se non si specifica un numero, tutti i buffer sono memorizzati.
viminfo-'
' Il numero massimo di file precedentemente aperti per cui vengono ricordati i marcatori. Questo parametro deve sempre venir incluso se 'viminfo' non è nullo.
Inserirlo significa anche che la |**jumplist**|e la |**changelist**| saranno immagazzinate nel file viminfo.
viminfo-/
/ Numero massimo di elementi nella storia ("history") delle stringhe di ricerca che verrà salvato. Se diverso da zero verranno salvate anche le precedenti espressioni usate per la ricerca e le relative stringhe di sostituzione.
Se non inserito viene usato il valore di 'history'.
viminfo-:
: Numero massimo di elementi nella storia ("history") delle righe di comando da salvare. Se non inserito viene usato il valore di 'history'.
viminfo-<
< Numero massimo di righe da salvare per ogni registro. Se a zero, i registri non sono salvati. Se omesso, tutte le righe sono salvate. '"' è il nome che aveva in precedenza questo elemento.
Vedere anche l'elemento 's' più sotto: il limite è specificato in Kbyte.
viminfo-@
@ Numero massimo di elementi nella storia ("history") della righe in input da salvare. Se non incluso, verrà usato il valore di 'history'.
viminfo-c
c Se inserito, converte il testo nel file viminfo dall''encoding' impiegato mentre si scriveva il file all''encoding' attualmente in uso. Vedere |**viminfo-encoding**|.
viminfo-f
f Se si debbono memorizzare i marcatori dei file. Se zero, i marcatori dei file (da '0 a '9, da 'A a 'Z) non vengono salvati. Se non presente o se diverso da zero, verranno salvati tutti. '0 viene usato per la posizione corrente del

cursore (quando uscite o eseguite il comando ":wviminfo").
viminfo-h
h Disabilita l'effetto di 'hlsearch' quando si carica il file viminfo. Se non inserito, l'evidenziazione dipende dal fatto che sia stato usato o meno ":nohlsearch" dopo l'ultimo comando di ricerca.
viminfo-n
n Nome del file viminfo. Il nome deve seguire immediatamente la lettera 'n'. Deve essere alla fine dell'opzione! Se l'opzione 'viminfofile' è impostata, il nome di file là specificato prevale su quello dato qui con l'opzione 'viminfo'. Le variabili d'ambiente vengono valutate quando viene aperto il file, non quando l'opzione viene impostata.
viminfo-r
r Supporti rimovibili. L'argomento è una stringa (sino alla ',' successiva). Questo parametro può venir specificato più volte. Ognuna specifica l'inizio di un percorso per il quale nessun marcatore verrà memorizzato. Ciò per evitare di memorizzare qualcosa di relativo ai supporti rimovibili. Per MS-DOS potreste impiegare "ra:,rb:", per Amiga "rdf0:,rdf1:,rdf2:". Potete anche usarlo per file temporanei, ad es., per Unix: "r/tmp". Non contano maiuscole o minuscole. La lunghezza massima di ciascun argomento 'r' è di cinquanta caratteri.
viminfo-s
s Massima dimensione di un elemento in Kbyte. Se è a zero, i registri non vengono salvati. Per ora riguarda solo i registri. Il valore di default "s10" esclude dal salvataggio i registri contenenti più di 10 Kbyte di testo. Vedere anche l'elemento '<' più sopra: limite al numero di linee.

Esempio: >
:set viminfo='50,<1000,s100,:0,n~/vim/viminfo
<
'50 Verranno memorizzati i marcatori degli ultimi cinquanta file che avete aperto.
<1000 I contenuti dei registri (fino a mille righe ciascuno) verranno memorizzati.
:0 La storia della riga di comando non verrà salvata.
n~/vim/viminfo Il nome del file da usare è "~/vim/viminfo".
no / Poiché non viene specificata '/' sarà impiegato il valore di default, che è, di salvare tutta la storia delle ricerche, e anche le precedenti espressioni di ricerca e sostituzione.
no % L'elenco dei buffer non verrà salvato e neppure riletto.
no h 'hlsearch' (l'evidenziazione delle stringhe di ricerca) verrà ripristinata.

Quando si imposta 'viminfo' a partire da un valore nullo, potete usare |**:rviminfo**| per caricare i contenuti del file, perché ciò non avviene automaticamente.

Quest'opzione non può venire impostata da |**modeline**|, o nel |**sandbox**|, per motivi di sicurezza.
NOTA: Quest'opzione è impostata al valore di default di Vim quando si imposta a off 'compatible'.

'viminfofile' *'vif'*
'viminfofile' 'vif' stringa (default: "")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+viminfo**|}
Se non consiste nella stringa nulla, prevale sul nome di file usato per viminfo.
Quando vale "NONE" non si legge né scrive alcun file viminfo.
Quest'opzione di può impostare con il flag |**-i**| dalla riga di comando.
Il flag |**--clean**| sulla riga di comando imposta l'opzione a "NONE".

'virtualedit' *'ve'*

'virtualedit' 've'	stringa (default: "")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+virtualedit**|}
Un elenco separato da virgole, di queste parole:
block	Consente la modifica virtuale nel modo Visual block.
insert	Consente la modifica virtuale nel modo Insert.
all	Consente la modifica virtuale in tutti i modi.
onemore	Consente al cursore di posizionarsi subito dopo la
fine della riga.

Modifica virtuale significa che il cursore può venir posizionato ove non ci sia attualmente alcun carattere. Ciò può essere in mezzo allo spazio vuoto "prodotto" da un <Tab> oppure dopo la fine della riga. Utile per selezionare un rettangolo in modo Visual e modificare una tabella.
"onemore" non è lo stesso, consentirà solo di spostare il cursore subito dopo l'ultimo carattere della riga. Questo rende più consistenti alcuni comandi. In precedenza il cursore era sempre oltre la fine della riga, se questa era una riga vuota. Ma questo modo di procedere non è affatto compatibile con Vi. Può anche rendere inoperanti alcuni "plugin" o script Vim. Per esempio perché |**l**| può muovere il cursore DOPO l'ultimo carattere. Usare con cautela!
Se si usa il comando `$`, il cursore va a finire sull'ultimo carattere della riga, e non oltre. Questo può risultare in uno spostamento del cursore a sinistra!
Il comando `g$` sposta il cursore alla fine della linea sullo schermo.
Non ha senso combinare "all" con "onemore", ma non viene inviato alcun messaggio di avvertimento se lo si fa.
NOTA: Quest'opzione è impostata a "" se si attiva 'compatible'.

'visualbell' *'vb'* *'novisualbell'* *'novb'* *beep*
'visualbell' 'vb'	booleana	(default: off)
globale
{non in Vi}
Usa una campanella visuale invece del suono. Il codice a terminale per mostrare la campanella visuale specificato con 't_vb'. Se non si vuole né un lampo né un suono si usi: >
:set vb t_vb=
<	Se si preferisce una campanella visuale breve, si può specificare come indicato qui in molti terminali: >
:set vb t_vb=¿[?5h$<100>¿[?5l
<	Qui $<100> specifica il tempo (in millisecondi). Si può usare un valore più piccolo o più grande per avere un campanello visuale più breve o più prolungato.

Nota: Vim limita il campanello a una volta ogni mezzo secondo. Ciò evita il tempo di attesa per la fine del campanello visuale quando ci sono tanti campanelli successivi, p.es. su un tasto ripetuto più volte. Ciò accade anche se non si specifica 'visualbell'.

Nella GUI, 't_vb' corrisponde a "<Esc>|f", che inverte lo schermo per 20 millisecondi. Se preferite una durata diversa impiegate "<Esc>|40f", dove 40 è il tempo espresso in millisecondi.

Nota: Quando la GUI parte, 't_vb' viene ricondotto al proprio valore di default. Potreste volerlo impostare nuovamente nel vostro |**gvimrc**|.

Non funziona in Amiga, dove si ottiene sempre un lampo dello schermo. Vedere anche 'errorbells'.

'warn' *'nowarn'*
'warn'	booleana	(default: on)
globale
Dà un messaggio di avvertimento se viene usato un comando di shell mentre il buffer è cambiato.

'weirdinvert' *'wiv'* *'noweirdinvert'* *'nowiv'*
'weirdinvert' 'wiv'	booleana	(default: off)
globale

{non in Vi}
Quest'opzione ha lo stesso effetto dell'opzione di terminale 't_xs'.
È disponibile per compatibilità all'indietro con la versione 4.x.
L'impostazione di 'weirdinvert' ha l'effetto di rendere 't_xs' non
vuoto e viceversa. Non ha alcun effetto se state usando la GUI.

'whichwrap' *'ww'*
'whichwrap' 'ww' stringa (default Vim: "b,s", default Vi: "")
globale
{non in Vi}
Consente ai tasti specificati che spostano il cursore a
destra/sinistra di spostarsi alla riga precedente/successiva quando
il cursore sia sul primo/ultimo carattere della riga. Specificare
i caratteri uno dopo l'altro [senza separatori] per permettere che
ciò avvenga per i seguenti tasti:

car.	tasto	modo ~
b	<BS>	Normal e Visual
s	<Space>	Normal e Visual
h	"h"	Normal e Visual (non raccomandato)
l	"l"	Normal e Visual (non raccomandato)
<	<Left>	Normal e Visual
>	<Right>	Normal e Visual
~	"~"	Normal
[	<Left>	Insert e Replace
]	<Right>	Insert e Replace

Ad esempio: >
:set ww=<,>,[,]
< consente di passare alla riga precedente/seguente solo se vengono
adoperati i tasti di cursore.
Quando i tasti di movimento vengono usati in combinazione con un
operatore di cancellazione o di modifica, anche la <EOL> [il carattere
di fine riga] viene contata come se fosse un carattere. Ciò rende
"3h" diverso da "3dh" se il cursore oltrepassa la fine della riga.
Questo vale anche per "x" e "X", perché fanno la stessa cosa di "dl" e
"dh". Usandolo potreste anche utilizzare la mappatura ":map <BS> X"
per ottenere la cancellazione, usando il backspace, del carattere
prima del cursore.
Se si aggiunge 'l' ed è usato dopo un operatore a fine riga, non si
sposterà sulla riga seguente. Questo permette a "dl", "cl", "yl",
etc. di funzionare normalmente.
NOTA: Quest'opzione è impostata al valore di default di Vi quando
'compatible' è a on oppure al valore di default di Vim quando
'compatible' è a off.

'wildchar' *'wc'*
'wildchar' 'wc' numero (default Vim: <Tab>, default Vi: CTRL-E)
globale
{non in Vi}
Il carattere da immettere per avviare la valutazione delle wildcard
(metacaratteri) da riga-comando, come specificato con 'wildmode'.
Il carattere non viene riconosciuto se usato entro una macro. Vedere
'wildcharm' per questo.
Trovate maggiori informazioni a: |**cmdline-completion**|.
Sebbene 'wc' sia un'opzione numerica, potete impostarla su di un tasto
speciale: >
:set wc=<Esc>
< NOTA: Quest'opzione è impostata al valore di default di Vi quando
'compatible' è a on oppure al valore di default di Vim quando
'compatible' è a off.

'wildcharm' *'wcm'*
'wildcharm' 'wcm' numero (default: nessuno (0))
globale
{non in Vi}
'wildcharm' funziona esattamente come 'wildchar', tranne che viene
riconosciuto se usato entro una macro. Potete trovare tasti "di
riserva" per la riga-comando adatti a quest'opzione guardando
in |**ex-edit-index**|. Normalmente non attiverete mai 'wildcharm' come
si impostano altre opzioni, va usato solo nelle mappature che
richiamano automaticamente il modo Completion, ad es.: >
:set wcm=<C-Z>
:cnoremap ss so $vim/sessions/*.vim<C-Z>
< In questo caso, dopo avere scritto :ss, potrete usare CTRL-P & CTRL-N.

'wildignore' *'wig'*
'wildignore' 'wig' stringa (default: "")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+wildignore**|}
Un elenco di espressioni regolari che designano dei file. Un file che
corrisponda a una di queste espressioni è ignorato nella valutazione di
|**wildcard**| , nel completamento di nomi di file o directory e influenza
i risultati di |**expand()**|, |**glob()**| e |**globpath()**|, a meno che un flag
sia impostato per disabilitare ciò.
Le espressioni vengono usate come con |**:autocmd**|, vedere
|**autocmd-patterns**|.
Vedere anche 'suffixes'.
Esempio: >
:set wildignore=*.o,*.obj
< L'impiego di |**:set+=**| e |**:set-=**| è preferibile per aggiungere o
togliere un'espressione dall'elenco. Ciò permette di evitare
problemi laddove una futura versione usasse un altro default.

'wildignorecase' *'wic'* *'nowildignorecase'* *'nowic'*
'wildignorecase' 'wic' booleana (default: off)
globale
{non in Vi}
Quando è impostata maiuscolo/minuscolo è ignorato nel completare nomi
di file e directory.
Non ha effetto quando 'fileignorecase' è impostata a on.
Non è utilizzato quando sia è la shell a venire usata per
la valutazione di espressioni regolari, il che succede in presenza di
caratteri speciali.

'wildmenu' *'wmnu'* *'nowildmenu'* *'nowmnu'*
'wildmenu' 'wmnu' booleana (default: off, impostata a on in
|**defaults.vim**|)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|**+wildmenu**|}
Se 'wildmenu' è abilitato, il completamento da riga-comando
funziona meglio. Premendo il tasto 'wildchar' (di solito <Tab>) per
richiedere il completamento, le corrispondenze possibili vengono
mostrate proprio sopra la riga-comando, con la prima di esse
evidenziata (sovrascrivendo la riga di status, se ne esiste una).
Tasti che mostrano la corrispondenza precedente/seguente, quali
<Tab> o CTRL-P/CTRL-N, causano il trasferimento dell'evidenziazione
sulla corrispondenza prescelta.
Quando si usa 'wildmode', il modo "wildmenu" si applica se è stato
specificato "full". "longest" e "list" non avviano il modo
"wildmenu".
Potete controllare il modo corrente con |**wildmenumode()**|.
Se ci fossero più corrispondenze di quelle che stanno su una riga,
un ">" apparirà sulla destra e/o un "<" sulla sinistra. La riga di
stato scorrerà secondo necessità.
Il modo "wildmenu" viene abbandonato premendo un tasto che non
seleziona un completamento.
Quando "wildmenu" è attivo i tasti che seguono hanno significati
speciali:

<Left> <Right> - sceglie la corrispondenza precedente/successiva
(come CTRL-P/CTRL-N)
<Down> - nel completamento del nome del nome_file/menù: si
sposta verso una sottodirectory o sotto-menù.
<CR> - nel completamento del menù, quando il cursore è
subito dopo un punto: si sposta verso un sotto-menù.
<Up> - nel completamento del nome del nome_file/menù: sale
verso la directory o il menù precedente.

Ciò rende i menù accessibili dalla console |**console-menus**|.

Se preferite usare i tasti <Left> e <Right> per spostare il cursore
invece che per scegliere fra le corrispondenze, usate questa
mappatura: >

```
                :cnoremap <Left> <Space><BS><Left>
                :cnoremap <Right> <Space><BS><Right>
<
```

L'evidenziazione del "WildMenu" viene usata per mostrare la corrispondenza corrente |**hl-WildMenu**|.

'wildmode' *'wim'*

'wildmode' 'wim' stringa (default Vim: "full")
globale
{non in Vi}

Il modo Completion viene usato per il carattere specificato con 'wildchar'. È un elenco separato da virgole che può avere fino a quattro parti. Ogni parte specifica cosa fare per ogni successivo uso di 'wildchar'. La prima parte definisce il comportamento per la prima utilizzazione di 'wildchar'. La seconda parte per la seconda utilizzazione e così via.
Ecco i valori possibili per ogni parte:

""	Effettua il completamento solo alla prima corrispondenza.
"full"	Completa la prossima corrispondenza esatta. Dopo l'ultima corrispondenza la stringa iniziale viene usata per cercare ancora la prima occorrenza seguente.
"longest"	Completa sino alla più lunga delle stringhe in comune. Se ciò non corrisponde a una stringa più lunga, utilizzare la prossima parte.
"longest:full"	Come "longest", ma avvia anche 'wildmenu' se abilitato.
"list"	Se c'è più di una corrispondenza, elencarle tutte.
"list:full"	Se c'è più di una corrispondenza, elencarle tutte e completare la prima.
"list:longest"	Se c'è più di una corrispondenza, elencarle tutte e effettuare il completamento sino alla stringa più lunga in comune.

Se c'è soltanto una corrispondenza essa viene completata in qualunque caso.

Ad es.: >

```
                :set wildmode=full
<
```

Completa la prima corrispondenza completa, la successiva e così via (questo è il default) >

```
                :set wildmode=longest,full
<
```

Completa la più lunga stringa in comune, quindi ogni corrispondenza completa >

```
                :set wildmode=list:full
<
```

Elenca ogni corrispondenza e completale tutte >

```
                :set wildmode=list,full
<
```

Elenca tutte le corrispondenze senza effettuare il completamento, poi lo effettua per tutte >

```
                :set wildmode=longest,list
<
```

Completa le stringhe più lunghe in comune, poi elenca le alternative. Trovate maggiori informazioni a: |**cmdline-completion**|.

'wildoptions' *'wop'*

'wildoptions' 'wop' stringa (default: "")
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+wildignore**|}

Una lista di parole che cambiano il modo in cui viene effettuato il completamento della riga-comandi.
Al momento è consentito specificare una sola parola:

tagfile	Quando si usa CTRL-D per elencare dei tag corrispondenti, il genere di tag e il file da cui proviene il tag sono elencati. Viene visualizzata una sola corrispondenza per riga. Tipi di tag usati spesso sono: d #define f funzione

Vedere anche |**cmdline-completion**|.

'winaltkeys' *'wak'*

'winaltkeys' 'wak' stringa (default: "menu")
globale
```
```

{non in Vi}
{usata soltanto nelle GUI Win32, Motif, GTK e Photon}
Alcune versioni di GUI consentono di accedere al menù usando il tasto ALT in combinazione con una lettera che appare sottolineata nel menù. Ciò è in conflitto con l'uso del tasto ALT per le mappature e per la creazione di caratteri speciali. Quest'opzione dice cosa fare:

no	Il tasto ALT non verrà usato per il menù. Le combinazioni del tasto ALT possono venir mappate ma non vengono eseguite automaticamente. Per farlo usare il comando |**:simalt**|.
yes	La gestione del tasto ALT viene fatta dalla GUI. Le combinazioni con il tasto ALT non possono venire mappate.
menù	Se si usa ALT in combinazione con un carattere che è valido per accedere al menù, questo verrà gestita dalla GUI. Gli altri tasti possono essere mappati.

Se il menù viene disabilitato escludendo 'm' da 'guioptions', il tasto ALT non è mai usato per il menù.
Quest'opzione non è usata con <F10>; in Win32 e con GTK <F10> selezionerà il menù, a meno che sia stata mappata.

'window' *'wi'*
'window' 'wi' numero (default: l'altezza dello schermo - 1)
globale
Altezza della finestra. Da non confondere con l'altezza della finestra di Vim, per la quale va usato 'lines'.
Usata per |**CTRL-F**| e |**CTRL-B**|, quando c'è solo una finestra e il valore è inferiore a 'lines' - 1. Lo schermo scorrerà 'window' - 2 righe, con un minimo di 1.
Quando 'window' è uguale a 'lines' - 1 CTRL-F e CTRL-B scorrono in una maniera molto più furba, tenendo conto delle righe che occupano più di una riga sullo schermo (wrapping).
Quando si cambiano le dimensioni della finestra Vim, se il valore è minore di 1 o maggiore o uguale a 'lines", sarà impostato a 'lines' -1.
{Vi usa l'opzione per specificare il numero di righe visualizzate}

'winheight' *'wh'* *E591*
'winheight' 'wh' numero (default: 1)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+windows**|}
Numero minimo di righe della finestra corrente. Non si tratta di un minimo assoluto, Vim userà meno righe se non c'è abbastanza spazio. Se si sta usando una finestra che è più piccola, la sua dimensione verrà aumentata a spese dell'altezza delle altre finestre.
Impostare 'winheight' a un numero basso per le operazioni normali. Impostarlo a 999 per rendere la finestra attiva sempre grande quasi come l'intero schermo. Le altre finestre saranno solo alte 'winminheight' righe. Ciò ha lo svantaggio che ":all" creerà solo due finestre. Per evitare che "vim -o 1 2 3 4" crei solo due finestre, impostate l'opzione solo dopo l'avvio di Vim, usando l'evento |**VimEnter**|: >
```
au VimEnter * set winheight=999
```
< Il valore minimo è 1.
L'altezza non viene più modificata dando dei comandi che modificano l'altezza della finestra corrente.
'winheight' agisce sulla finestra corrente. Usate 'winminheight' per impostare l'altezza minima di altre finestre.

'winfixheight' *'wfh'* *'nowinfixheight'* *'nowfh'*
'winfixheight' 'wfh' booleana (default: off)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità |**+windows**|}
Mantenere l'altezza della finestra quando ne vengono aperte o chiuse altre e l'opzione 'equalalways' sia impostata. Ciò vale anche per |**CTRL-W_=**|. Impostata per default per la finestre |**preview-window**| e |**quickfix-window**|.
L'altezza può comunque essere cambiata quando si ecceda lo spazio disponibile.

'winfixwidth' *'wfw'* *'nowinfixwidth'* *'nowfw'*
'winfixwidth' 'wfw' booleana (default: off)
locale alla finestra
{non in Vi}
{non disponibile se compilato senza la funzionalità
|+windows|}
Mantenere la larghezza della finestra quando si aprono o chiudono finestre e 'equalalways' è impostato. Vale anche per |CTRL-W_=|.
La larghezza può comunque essere cambiata quando si ecceda lo spazio disponibile.

'winminheight' *'wmh'*
'winminheight' 'wmh' numero (default: 1)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|+windows|}
L'altezza minima di una finestra, diversa da quella corrente.
Si tratta di un minimo assoluto. La finestra non potrà mai diventare più piccola.
Se impostato a zero, le finestre potrebbero venir "schiacciate" a zero righe (solo la barra di stato) se necessario. Torneranno al minimo di una sola riga quando diverranno correnti (poiché il cursore deve avere una riga su cui stare).
Usare 'winheight' per impostare un'altezza minima per la finestra corrente.
Quest'opzione viene controllata soltanto quando si rende una finestra più piccola. Non usare un numero troppo elevato, potrebbe generare degli se si aprono tante finestre. Un valore fra 0 e 3 è ragionevole.

'winminwidth' *'wmw'*
'winminwidth' 'wmw' numero (default: 1)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|+vertsplit|}
La larghezza minima di una finestra, diversa da quella corrente.
Si tratta di un minimo assoluto. La finestra non potrà mai diventare più piccola.
Se impostato a zero, le finestre potrebbero venir "schiacciate" a zero colonne (solo la barra di stato) se necessario. Torneranno al minimo di una sola riga quando diverranno correnti (poiché il cursore deve avere una riga su cui stare).
Usate 'winwidth' per impostare una larghezza minima per la finestra corrente.
Quest'opzione viene controllata soltanto quando si rende una finestra più piccola. Non usare un numero troppo elevato, potrebbe generare degli se si aprono tante finestre. Un valore fra 0 e 12 è ragionevole.

'winptydll'
'winptydll' stringa (default: "winpty32.dll" o "winpty64.dll")
globale
{non in Vi}
{disponibile solo se compilato con la funzionalità
|terminal| in MS-Windows}
Specifica il nome della libreria condivisa winpty shared, usata per il comando |:terminal|. Il valore di default dipende da come è stato generato Vim, (se a 32-bit o a 64-bit). Se la libreria non viene trovata, si tenta di trovare "winpty.dll" come ulteriore tentativo.
Le variabili d'ambiente sono valutate |:set_env|.
Quest'opzione non può essere impostata da |modeline|, o nel |sandbox|, per motivi di sicurezza.

'winwidth' *'wiw'* *E592*
'winwidth' 'wiw' numero (default: 20)
globale
{non in Vi}
{non disponibile se compilato senza la funzionalità
|+vertsplit|}
Numero minimo di colonne per la finestra corrente. Non si tratta di un minimo assoluto, Vim userà meno colonne se non c'è abbastanza

spazio.
Se la finestra corrente è più piccola la sua larghezza verrà aumentata, a spese della larghezza delle altre finestre. Impostarlo a 999 per rendere la finestra corrente grande come l'intero schermo. Usare un numero minore per le operazioni normali.
La larghezza non viene più modificata dando dei comandi che modificano la larghezza della finestra corrente.
'winwidth' agisce sulla finestra corrente. Usate 'winminwidth' per impostare la larghezza minima di altre finestre.

'wrap' *'nowrap'*
'wrap' booleana (default: on)
locale alla finestra
{non in Vi}
Quest'opzione cambia il modo di visualizzazione del testo. Non modifica il testo nel buffer, vedere 'textwidth' per questo.
Se impostata a on, le righe più lunghe della larghezza della finestra verranno continuate nella riga successiva della finestra.
Quando è impostata a off le righe più lunghe della larghezza della finestra vengono visualizzate solo per la parte che può essere contenuta nella finestra. Quando il cursore viene spostato verso la parte non visualizzata, lo schermo scorrerà orizzontalmente.
Le riga verrà interrotta a metà di una parola se necessario. Vedere 'linebreak' per ottenere l'interruzione alla fine di una parola.
Per rendere un po' più utile lo scorrimento orizzontale provate questo: >
```
:set sidescroll=5
:set listchars+=precedes:<,extends:>
```
< Vedere 'sidescroll', 'listchars' e |**wrap-off**|.
Quest'opzione non può essere impostata da |**modeline**| se l'opzione 'diff' è attiva.

'wrapmargin' *'wm'*
'wrapmargin' 'wm' numero (default: 0)
locale al buffer
Numero di caratteri dal bordo destro della finestra ove inizia l'interruzione di riga. Se si scrive del testo oltre questo limite viene inserito un <EOL> e l'immissione continua sulla riga successiva.
Opzioni che aggiungono un margine, come 'number' e 'foldcolumn' riducono ulteriormente la larghezza del testo. Questo comportamento è compatibile con Vi.
Se 'textwidth' è diverso da zero quest'opzione non viene usata.
Quest'opzione è messa a off quando si imposta l'opzione 'paste' e ripristinata quando l'opzione 'paste' è messa a off.
Vedere anche 'formatoptions' e |**ins-textwidth**|.
{Vi: funziona in modo differente e meno utile}

'wrapscan' *'ws'* *'nowrapscan'* *'nows'*
'wrapscan' 'ws' booleana (default: on) *E384* *E385*
globale
La ricerca prosegue oltre la fine del file. Vale anche per i comandi |**]s**| e |**[s**|, che effettuano la ricerca degli errori ortografici.

'write' *'nowrite'*
'write' booleana (default: on)
globale
{non in Vi}
Consente di salvare un file. Se è impostata a off, non si possono salvare i file.
Può essere usata per il modo sola lettura, ove le modifiche al testo non sono ancora ammesse. Può venir inattivata con gli argomenti a riga di comando|**-m**| o |**-M**|. Il filtraggio del testo (passandolo a qualche comando esterno) è ancora possibile, sebbene ciò richieda di scrivere un file temporaneo.

'writeany' *'wa'* *'nowriteany'* *'nowa'*
'writeany' 'wa' booleana (default: off)
globale
Consente di salvare qualunque file senza che sia necessario il "!" per forzare la riscrittura.

'writebackup' *'wb'* *'nowritebackup'* *'nowb'*

'writebackup' 'wb'	booleana	(attiva per default con l'opzione
|**+writebackup**| impostata, altrimenti inattiva)
globale
{non in Vi}

Fare un backup prima di sovrascrivere un file. Tale copia viene cancellata dopo che il file è stato salvato, a meno che l'opzione 'backup' sia anch'essa attiva.
ATTENZIONE: Disattivare quest'opzione implica che se Vim non riesce a scrivere il vostro buffer correttamente e in seguito, per una ragione qualunque, la sessione Vim termina, vanno persi sia il file originale che quello che stavate scrivendo. Conviene inattivare quest'opzione solo se il vostro disco è quasi pieno, e questo causa la mancata riscrittura.
Vedere |**backup-table**| per un'altra spiegazione.
Quando il nome di file corrisponde a uno di quelli specificati con l'opzione 'backupskip' non viene effettuata alcuna copia.
NOTA: Quest'opzione viene impostata al valore di default se si imposta 'compatible'.

'writedelay' *'wd'*

'writedelay' 'wd'	numero (default: 0)
globale
{non in Vi}

Il numero di millisecondi di attesa prima di inviare ogni carattere allo schermo. Quando il valore è diverso da zero i caratteri vengono inviati al terminale uno per volta. Per il terminale MS-DOS "pcterm" questo non funziona. Da utilizzare per il debug.

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

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