

usr_06.txt Per Vim version 7.0. Ultima modifica: 2006 Apr 24

VIM USER MANUAL - di Bram Moolenaar
Traduzione di questo capitolo: Alessandro Melillo

Usare l'evidenziazione della sintassi

Il testo in bianco e nero è noioso. Col colore, il vostro file prende vita. E non solo ha un bell'aspetto, ma velocizza anche il vostro lavoro. Cambiate i colori utilizzati per i diversi tipi di testo. Stampate i vostri testi, con i colori che vedete a schermo.

06.1	Abilitare l'evidenziazione
06.2	Nessun colore o colori sbagliati?
06.3	Colori diversi
06.4	Con o senza i colori
06.5	Stampare a colori
06.6	Ulteriori letture

Capitolo seguente:	usr_07.txt	Elaborare più di un file
Capitolo precedente:	usr_05.txt	Configurazioni personali
Indice:	usr_toc.txt	

=====

06.1 Abilitare l'evidenziazione

Basta un semplice comando:

```
:syntax enable
```

Dovrebbe funzionare nella maggior parte dei casi avere per avere i colori nei vostri file. Vim individuerà automaticamente il tipo di file e caricherà la giusta evidenziazione della sintassi. I commenti diventeranno blu, le parole chiave marroni e le stringhe rosse. Questo rende semplice revisionare il file. Dopo un pò vi renderete conto che il testo in bianco e nero vi rallenta!

Se volete utilizzare sempre l'evidenziazione della sintassi, mettete il comando ":syntax enable" nel vostro |vimrc|.

Se volete utilizzarlo solo quando il terminale supporta il colore, potete inserire in |vimrc| : >

```
if &t_Co > 1
    syntax enable
endif
```

Se volete l'evidenziazione solo nella versione GUI, mettete il comando "syntax enable" nel vostro |gvimrc| .

=====

06.2 Nessun colore o colori sbagliati?

I motivi possono essere diversi:

- Il vostro terminale non supporta il colore.
Vim utilizzerà il grassetto, il corsivo e il sottolineato, anche se il risultato non ha un bell'aspetto. Probabilmente vorrete provare un terminale col colore. Per Unix, raccomando xterm del progetto XFree86: |xfree-xterm|.

- Il vostro terminale supporta il colore, ma Vim non lo sa.
Assicuratevi del corretto settaggio di \$TERM. Per esempio, usando un xterm che supporta il colore: >

```
setenv TERM xterm-color
```

<

```
o (a secondo della shell): >
```

```
TERM=xterm-color; export TERM
```

<

Il nome del terminale deve corrispondere a quello che state usando. Se ancora non funziona, date un'occhiata a |xterm-color|, che illustra alcuni modi di far sì che VIM mostri i colori (non solo per un xterm).

- Il tipo di file non viene riconosciuto.

In fondo, Vim non conosce tutti i tipi di file e talvolta è quasi impossibile dire quale sia il linguaggio del file. Provate questo comando: >

```
:set filetype
```

<

Se il risultato è "filetype=" allora il problema è senza dubbio che Vim non sa che tipo di file sia. Potete impostare il tipo file manualmente: >

```
:set filetype=fortran
```

<

Per vedere quali tipi sono disponibili, guardate nella directory \$VIMRUNTIME/syntax. Per la GUI potete usare il menu Sintassi. Settare il tipo di file può essere fatto anche con una `|modeline|`, in modo che il file venga evidenziato ogni volta che lo modificate. Per esempio, questa riga può essere usata in un Makefile (mettetela vicino all'inizio o alla fine del file): >

```
# vim: syntax=make
```

<

Potreste riuscire a determinare da soli il tipo di file. Spesso si può usare l'estensione (dopo il punto). Vedete `|new-filetype|` per sapere come dire a Vim di individuare il tipo di file.

- Non è definita una evidenziazione per il vostro tipo di file. Potreste provare ad usare un tipo simile impostandolo manualmente come spiegato sopra. Se il risultato non fosse soddisfacente, potreste scrivervi il vostro file di sintassi. Vedere `|mysyntaxfile|`.

Oppure i colori potrebbero essere sbagliati:

- Il testo colorato è molto difficile da leggere. Vim individua il colore del vostro sfondo. Se è nero (o un altro colore scuro) utilizza colori chiari per il testo. Se è bianco (o un altro colore chiaro) utilizza colori scuri per il testo. Se Vim sbaglia a determinare il colore di sfondo, il testo sarà difficile da leggere. Per risolvere questo inconveniente, impostate l'opzione `'background'`. Per uno sfondo scuro: >

```
:set background=dark
```

<

E per uno sfondo chiaro: >

```
:set background=light
```

<

Assicuratevi di impostarlo prima del comando `":syntax enable"`, altrimenti i colori saranno già stati impostati. Potreste fare un `":syntax reset"` dopo aver impostato `"background"` per far sì che Vim reimposti i colori.

- I colori sono sbagliati quando scorrete dal basso all'alto. Vim non legge l'intero file per scansionare il testo. Inizia a scansionarlo in qualsiasi punto vi troviate. Questo fa risparmiare un mucchio di tempo, ma a volte i colori risultano sbagliati. Una semplice correzione è premere `CTRL-L`. O scorrere un pò indietro e poi ancora un pò avanti. Per una correzione seria, vedete `|:syn-sync|`. Alcuni file di sintassi hanno un modo per far sì che guardi ben più indietro, vedete l'help dello specifico file di sintassi. Per esempio, `|tex.vim|` per la sintassi TeX.

=====

06.3 Colori Diversi

```
*:syn-default-override*
```

Se non vi piacciono i colori di default, potete scegliere un altro schema. Nella GUI usate il menu Edit/Color Scheme. Potete anche scrivere il comando: >

```
:colorscheme evening
```

"evening" è il nome dello schema. Ce ne sono diversi altri che potreste voler provare. Guardate nella directory \$VIMRUNTIME/colors.

Quando avete trovato lo schema che vi piace aggiungete il comando `":colorscheme"` al vostro vimrc.

Potrete anche scrivere da soli il vostro schema. Ecco come:

1. Selezionate uno schema che ci si avvicina. Copiate questo file nella vostra directory di Vim. Per Unix, dovrebbe bastare questo: >

```
!mkdir ~/.vim/colors
!cp $VIMRUNTIME/colors/morning.vim ~/.vim/colors/mine.vim
```

<

Questo si fa da Vim, visto che conosce il valore di \$VIMRUNTIME.

2. Modificate il file dello schema. Queste voci sono utili:

term	attributi in un terminale in bianco e nero
cterm	attributi in un terminale a colori
ctermfg	colore di primo piano in un terminale a colori
ctermbg	colore di sfondo in un terminale a colori
gui	attributi nella GUI
guifg	colore di primo piano nella GUI
guibg	colore di sfondo nella GUI

Per esempio, per rendere verdi i commenti: >

```
:highlight Comment ctermfg=green guifg=green
```

<

Gli attributi utilizzabili per "cterm" e "gui" sono "bold" e "underline". Se li volete entrambi, usate "bold,underline". Per maggiori dettagli, vedete il comando |:highlight|.

3. Dite a Vim di usare il vostro schema di colori. Mettete questa riga nel vostro |vimrc|: >

```
colorscheme mine
```

Se volete vedere come appaiono le combinazioni di colori più usate, utilizzate questo comando: >

```
:runtime syntax/colortest.vim
```

Vedrete del testo in varie combinazioni di colori. Potete quindi scegliere quelle più leggibili e carine.

```
=====
*06.4* Con o senza i colori
```

Mostrare il testo a colori impiega molte risorse. Se trovate che sia troppo lento, potete disabilitare l'evidenziazione per un istante: >

```
:syntax clear
```

Quando modificate un altro file (o lo stesso) i colori torneranno.

```
*:syn-off*
```

Se volete bloccare definitivamente l'evidenziazione usate: >

```
:syntax off
```

Questo disabiliterà completamente l'evidenziazione della sintassi e la rimuoverà immediatamente da tutti i buffer.

```
*:syn-manual*
```

Se volete abilitare l'evidenziazione solo per delle determinate righe, usate questo: >

```
:syntax manual
```

Questo abiliterà l'evidenziazione della sintassi, ma non la userà automaticamente quando si inizia a modificare un buffer. Per attivarla nel buffer corrente, impostate l'opzione 'syntax': >

```
:set syntax=ON
```

<

```
=====
*06.5* Stampare con i colori
```

```
*syntax-printing*
```

Nella versione per MS-Windows potete stampare il file corrente con questo

comando: >

```
:hardcopy
```

Otterrete la solita finestra di dialogo della stampante, dove potrete selezionare la stampante e alcune impostazioni. Se avete una stampante a colori, l'output su carta dovrebbe risultare lo stesso di quello a schermo con Vim. Ma se utilizzate uno sfondo scuro i colori verranno modificati per avere un buon aspetto su carta bianca.

Ci sono diverse opzioni che possono cambiare il modo in cui Vim stampa:

```
'printdevice'  
'printhead'  
'printfont'  
'printoptions'
```

Per stampare solo un intervallo di righe, utilizzate la modalità Visual per selezionare le linee e poi usate il comando: >

```
v100j:hardcopy
```

"v" inizializza il Visual mode. "100j" si sposta in basso di 100 righe, evidenziandole. Infine ":hardcopy" le stampa. Ovviamente potete usare altri comandi per spostarvi in Visual mode.

Questo funziona anche sotto Unix se avete una stampante PostScript. Altrimenti c'è bisogno di un pò più di lavoro. Dovete prima convertire il testo in HTML, e poi stamparlo da un browser come Netscape.

Convertite il file corrente in HTML con questo comando: >

```
:source $VIMRUNTIME/syntax/2html.vim
```

Sentirete che sta macinando dati, e potrebbe essere necessario molto tempo per un file di grandi dimensioni.

Poco dopo, la finestra mostrerà il codice HTML. Adesso salvatelo da qualche parte (non importa dove, dopo lo cancellerete):

>

```
:write main.c.html
```

Aprirete questo file nel vostro browser preferito e stampatelo da lì. Se tutto va bene, l'output dovrebbe avere lo stesso aspetto che ha dentro Vim. Vedete |2html.vim| per i dettagli. Non dimenticate di cancellare l'HTML quando avete finito.

Invece di stampare, potreste anche mettere il file HTML su un web server, e lasciare che altri vedano il testo a colori.

=====

06.6 Ulteriori letture

usr_44.txt	Evidenziazione della vostra sintassi
syntax	Tutti i dettagli.

=====

Capitolo seguente: |usr_07.txt| Elaborare più di un file

Copyright: vedere |manual-copyright| vim:tw=78:ts=8:ft=help:norl:

Segnalare refusi a Bartolomeo Ravera - E-mail: barrav at libero.it
oppure ad Antonio Colombo - E-mail: azc100 at gmail.com