

손에 잡히는 vim (2/4)

김선영

[sunzero@gmail\(dot\)com](mailto:sunzero@gmail(dot)com)

버 전: 2014-10

인사이트 출판사 <http://blog.insightbook.co.kr>

가메 출판사 <http://www.kame.co.kr>

저자홈페이지 <http://sunzero.tistory.com>

Ch3. 옵션, 도움말, 에러처리



RTFM(Read The Fine Manual) - 인터넷격언

명령행 모드: 옵션 설정

❖ `~/.vimrc` 파일에 저장가능 (MS 윈도 버전의 경우 `~/_vimrc`)

▶ `:set name[=value]`

▶ `.vimrc` 에 지정할때는 `:` 은 필요없음

명령어	설 명
:set	현재 옵션 설정을 보여줌
:set all	모든 옵션 설정을 보여줌 (default 옵션까지 출력)
:set [no]name	name 에 해당하는 옵션을 on/off 함 (앞에 prefix 를 no 를 지정하는 경우가 off)
:set name!	name 옵션의 on, off 를 toggle 함
:set name=value	name 옵션에 value 의 값을 할당함

명령행모드: 옵션설정 (con't)

❖ :set 으로 현재 설정된 옵션 확인

```
~  
:set  
--- Options ---  
cscopetag          hlsearch          ttymouse=xterm2      t_Sf=^[[3%dm  
cscopeverbose      ruler              viminfo='20,"50  
helplang=en        scroll=5            window=11  
history=50          ttyfast          t_Sb=^[[4%dm  
backspace=indent,eol,start  
cscopeprg=/usr/bin/cscope  
fileencodings=ucs-bom,utf-8,latin1  
guicursor=n-v-c:block,o:hor50,i-ci:hor15,r-cr:hor30,sm:block,a:blinkon0  
Press ENTER or type command to continue
```

명령행 모드: 옵션 설정 (con't)

❖ :set all 로 현재 모든 옵션 확인

```
:set all
--- Options ---
  aleph=224                foldcolumn=0          nonumber                switchbuf=
noarabic                  foldenable            numberwidth=4          synmaxcol=3000
  arabicshape              foldexpr=0            omnifunc=               syntax=
noallowrevins             foldignore=#          operatorfunc=           tabline=
noaltkeymap               foldlevel=0           nopaste                 tabpagemax=10
  ambiwidth=single        foldlevelstart=-1     pastetoggle=            tabstop=8
noautochdir               foldmethod=manual     patchexpr=              tagbsearch
noautoindent              foldminlines=1        patchmode=              taglength=0
noautoread                foldnestmax=20        nopreserveindent        tagrelative
noautowrite               formatexpr=            previewheight=12        tagstack
noautowriteall            formatoptions=tcq     nopreviewwindow         term=xterm
-- More --
```

명령행 모드: 옵션 설정 (con't)

❖ :set nu

- ▶ 행번호를 표시해주는 옵션

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main()
4 {
5     printf(" Hello World \n");
6     exit(EXIT_FAILURE);
7 }
```

~
~
~

:set nu

7,1

All

명령행 모드: 옵션 설정 (con't)

❖ 주로 사용하는 옵션

옵션	설 명 (괄호 안은 full name)
nu	(number) 화면에 행번호 표시
ai	(autoindent) 자동 들여쓰기
cindent	C 언어 스타일의 들여쓰기
ts=value	(tabstop) 화면에 표시 될 탭 크기를 value 로 지정
sw=value	(shiftwidth) 자동 블록 이동시 열의 너비
fencs=value	(fileencodings) 지원할 파일 인코딩 방식 리스트 * 복수개 지정시 콤마로 구분합니다(공백 없음)
fenc=value	(fileencoding) 현재 파일 인코딩 방식을 지정

명령행 모드: 옵션 설정 (con't)

❖ .vimrc 예시

```
" 이것은 주석문이다 :-)  
set ai cindent  
set ts=4 sw=4  
set nu  
" euc-kr로 인코딩 된 파일을 읽기 위한 설정 (뒤에 파일 인코딩에서 다룸)  
set fencs=ucs-bom,utf-8,korea
```

- ▶ set fencs=ucs-bom,utf-8,euc-kr 로 사용하는 경우도 있다.
 - korea는 *nix 계열에서는 euc-kr, 윈도우 계열에서는 cp949로 바뀐다.
 - set fencs=ucs-bom,korea,utf-8 로 설정하면 euc-kr, cp949의 우선권이 높아진다. 따라서 새로운 파일은 euc-kr이나 cp949로 저장된다.

Tip!

- ❖ 파일별로 옵션을 오버라이딩 하고 싶을 경우

```
/* vim: set ts=2 sw=2: */  
#include <stdio.h>  
int main()  
{  
    printf("Hello world\n");  
    return 0;  
}
```

colorscheme

❖ color theme를 지원

- ▶ option이 아니므로 set 접두어를 쓰지 않는다.

```
:colorscheme
```

여기서 <Tab> 키를 누를때마다 한개씩
사용가능한 테마가 나타납니다.

<CTRL-D>로 탐색 가능

<pre>:colo</pre>					
blue	delek	evening	murphy	ron	torte
darkblue	desert	koehler	pablo	shine	zellner
default	elflord	morning	peachpuff	slate	
<pre>:colo</pre>					

- colorscheme의 약어인 **colo** 까지만 입력해도 된다.
- .vimrc에 등록해두면 매번 자동으로 설정될 수 있다.

colorscheme : ex

❖ colorscheme 예

```
#include <stdio.h> /* peachpuff */
#include <unistd.h>
#include <sys/types.h>
int main(void) {
    dprintf(STDOUT_FILENO, "Hello world (%d)\n", getpid());
    return 234;
}
```

```
#include <stdio.h> /* Slate */
#include <unistd.h>
#include <sys/types.h>
int main(void) {
    dprintf(STDOUT_FILENO, "Hello world (%d)\n", getpid());
    return 234;
}
```

다양한 컬러스키마를
볼 수 있다.

```
#include <stdio.h> /* desert */
#include <unistd.h>
#include <sys/types.h>
int main(void) {
    dprintf(STDOUT_FILENO, "Hello world (%d)\n", getpid());
    return 234;
}
```

help

❖ vim은 online help를 지원한다.

```
~  
:help word  
word                :<cword>  
word-count          'cinwords'  
word-motions        'lispwords'  
spell-wordlist-format 'iskeyword'  
internal-wordlist   'keywordprg'  
WORD                :syn-keyword  
ada#Word()          compl-keyword  
g:ada#WordRegex     spell-midword  
spell-WORDCHARS     spellbadword()  
:help word
```

<CTRL-D>으로
탐색을 한 결과

help : prefix

❖ 도움말에 사용되는 접두어

모드	접두어	예
일반	없음	:help x
입력	i_	:help i_CTRL-N
명령행	:	:help :w
비주얼	v_	:help v_u
vim 실행인수	-	:help -r
옵션	'	:help 'tabstop'
명령행 모드 특수키	c_	:help c_CTRL-B

help (con't)

❖ :help cindent

`cindent({lnum})`

`*cindent()`

Get the amount of indent for line `{lnum}` according to the C indenting rules, as with `'cindent'`.

The indent is counted in spaces, the value of `'tabstop'` is relevant. `{lnum}` is used just like in `getline`.

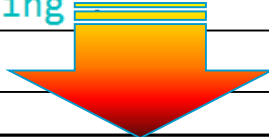
When `{lnum}` is invalid or Vim feature, -1 is returned.

See `C-indenting`

`+cindent`

tabstop 에서

^] 를 누르면...



`*'tabstop' - 'ts'`

`'tabstop' 'ts'`

number (default 8)

local to buffer

Number of spaces that a `<Tab>` in the file counts for. Also see `:retab` command, and `'softtabstop'` option.

Note: Setting `'tabstop'` to any other value than 8 can make your file appear wrong in many places (e.g., when printing it).

help (con't)

❖ ^{^]}

- ▶ Jump to the definition of the keyword under the cursor.

❖ ^{^T}

- ▶ Jump to older entry in the tag stack.

❖ :tags

- ▶ Show the contents of the tag stack.
The active entry is marked with a '>'.

❖ ^{^W^W}

- ▶ move cursor to window below/right of the current one.

vim error

❖ 파일을 중복해서 오픈 한 경우

E325: ATTENTION

Found a swap file by the name ".helloworld.c.swp"

owned by: linuxer dated: Sun Oct 31 04:10:33 2010

file name: ~linuxer/hello

modified: no

user name: linuxer host name: supermar.org

process ID: 28779 (still running)

While opening file "helloworld.c"

dated: Sun Oct 31 04:10:33 2010

(1) Another program may be editing the same file.

If this is the case, be careful not to end up with two different instances of the same file when making changes. Quit, or continue with caution.

(2) An edit session for this file crashed.

If this is the case, use ":recover" or "vim -r helloworld.c" to recover the changes (see ":help recovery").

If you did this already, delete the swap file ".helloworld.c.swp" to avoid this message.

Swap file ".helloworld.c.swp" already exists!

[O]pen Read-Only, (E)dit anyway, (R)ecover, (Q)uit, (A)bort:

다른 vim 이 열고
있음을 보여주고 있음

O, E, R, Q, A 의 다섯가지 메뉴

vim error (con't)

❖ vim crash로 swapfile이 제거되지 못한 경우

E325: ATTENTION

Found a swap file by the name ".helloworld.c.swp"

owned by: linuxer dated: Tue Nov 2 04:32:49 2010

file name: ~linuxer/helloworld.c

modified: no

user name: linuxer host name: dev.optimar.org

process ID: 28779

While opening file "helloworld.c"

dated: Sun Oct 31 04:10:33 2010

(1) Another program may be editing the same file.

If this is the case, be careful not to end up with two different instances of the same file when making changes.

Quit, or continue with caution.

(2) An edit session for this file crashed.

If this is the case, use ":recover" or "vim -r helloworld.c" to recover the changes (see ":help recovery").

If you did this already, delete the swap file ".helloworld.c.swp" to avoid this message.

Swap file ".helloworld.c.swp" already exists!

[O]pen Read-Only, (E)dit anyway, (R)ecover, (D)elete it, (Q)uit, (A)bort:

Ch4.문자열 관련 기능



머리가 나쁘면 손발이 고생한다. - 중국속담

Formatting

❖ center, right, left

- ▶ width를 기준으로 정렬을 한다.

**width=80의
center 결과**

What Is Vim?

Vim is a highly configurable text editor built to enable efficient text editing.
It is an improved version of the vi editor distributed with most UNIX systems.

Vim is often called a "programmer's editor," and so useful for programming that many consider it an entire IDE. It's not just for programmers, though. Vim is perfect for all kinds of text editing, editing configuration files.

:center

**:center 50 의
실행 결과**

1,11-35

Top

What Is Vim?

Vim is a highly configurable text editor built to enable efficient text editing.
It is an improved version of the vi editor distributed with most UNIX systems.

Vim is often called a "programmer's editor," and so useful for programming that many consider it an entire IDE. It's not just for programmers, though. Vim is perfect for all kinds of text editing, from composing email to editing configuration files.

:center 50

1,8-20

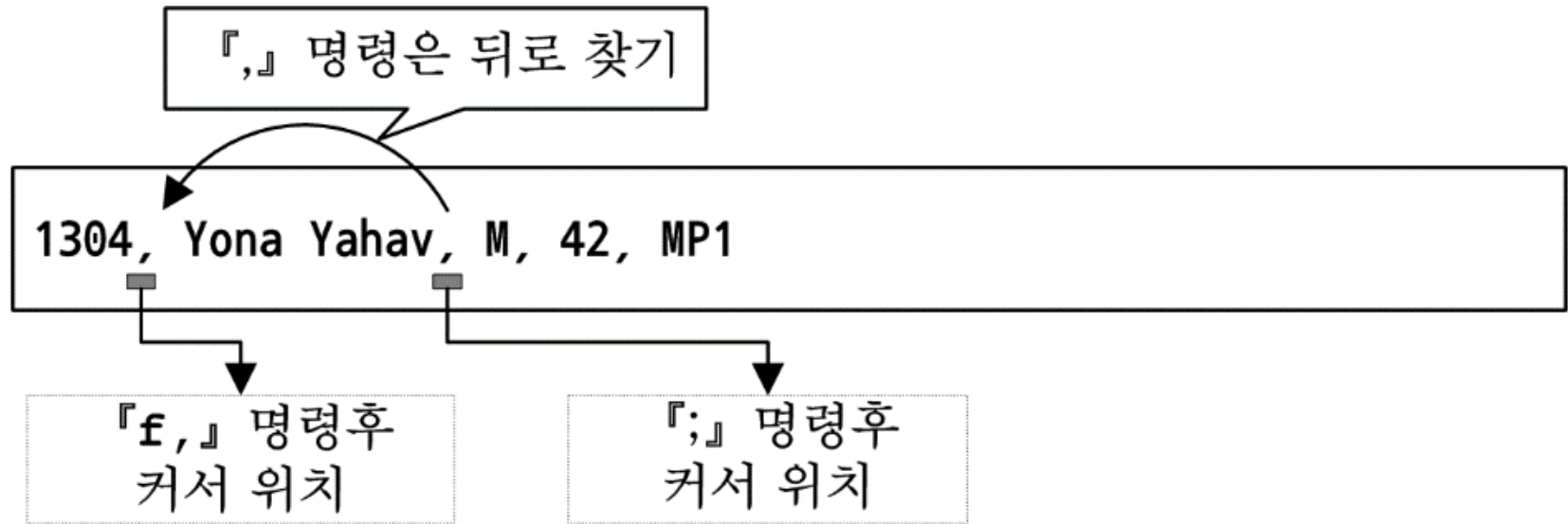
Top

find

❖ 문자 1개를 검색

명령어	설명
fc	문자 c 를 전방 검색
Fc	문자 c 를 후방 검색
;	최근 검색을 재검색
,	최근 검색을 반대방향으로 재검색

find (con't)



일반모드: 검색

❖ 명령어

- ▶ **/string : string**을 전방탐색 함
- ▶ **?string : string**을 후방탐색 함
- ▶ ***** : 현재 커서에 위치한 단어를 전방탐색 함
- ▶ **#** : 현재 커서에 위치한 단어를 후방탐색 함
- ▶ **n** : 다음 탐색 결과를 찾아냄
 - ↘ (전방탐색의 경우는 전방으로, 후방탐색의 경우는 후방으로 이동)
- ▶ **N** : n과 반대방향으로 다음 탐색
- ▶ **%** : 괄호의 짝을 찾아줌

일반모드: 검색 (con't)

❖ case-sensitive searching

- ▶ \c 접두어를 사용하거나...
- ▶ **ignorecase option**을 켜거나... (**option**은 뒤에서 다룹니다.)

```
/\cinter
```

대소문자를 구별하지 않고,
inter를 검색한다.

- ▶ \C는 \c의 반대 기능 = ignorecase가 켜진 경우에, case-sensitive searching을 가능

일반모드 : 검색 : REGEX

The capacity to endure a more or less monotonous life is one which should be acquired in childhood. Modern parents are greatly to blame in this respect; they provide their children with far too many passive amusements, such as shows and good things to eat, and they do not realize the importance to a child of having one day like another, except, of course, for somewhat rare occasions. - Bertrand Russell

/[a-g]re

2,55

[Top](#)



Tip!

❖ 검색 후 밝게 표시된 단어를 해제하고 싶을 때

▶ **:nohl**

↳ 일회성 highlightsearch 해제

▶ **:set nohls**

↳ highlightsearch 옵션 해제

교체



문자열 교체(substtue) 명령어

특수 문자 교체와 파일 형식(fileformat) 명령어

substitute

❖ 교체 : sed의 기능이 import 된 것 (**sed의 문법과 동일하다**)

▶ :[range]s/<찾는 문자열>/<교체할 문자열>/<옵션>

↳ separator로 slash(/)가 주로 사용되지만 다른 문자를 사용해도 무방하다. (예 : 콤마)

명령어 옵션	설 명
g	(global) 검색된 문자열 모두를 교체 * g 옵션 생략시 처음 1 개만 교체함
i	(ignore case) 대소문자 무시
c	(confirm) 교체 할 때마다 Yes/No 확인
e	교체 과정 중 에러 무시

substitute (con't)

❖ separator의 escape

- ▶ /home/linuxer 를 /home/unixer로 교체 하고자 한다면?

```
:%s//home/linuxer//home/unixer/g
```



```
:%s/\\/home\\/linuxer\\/\\/home\\/unixer/g
```



정답이지만 복잡하다!

```
:%s,/home/linuxer,/home/unixer,g
```



separator를 comma로 바꾸면?

substitute : fileformat : DOS/UNIX

❖ 특수 문자(예: NewLine)의 교체

형식	개행 문자
DOS/Windows	CR+LF
UNIX	LF

▶ CR의 입력 : ^V^M

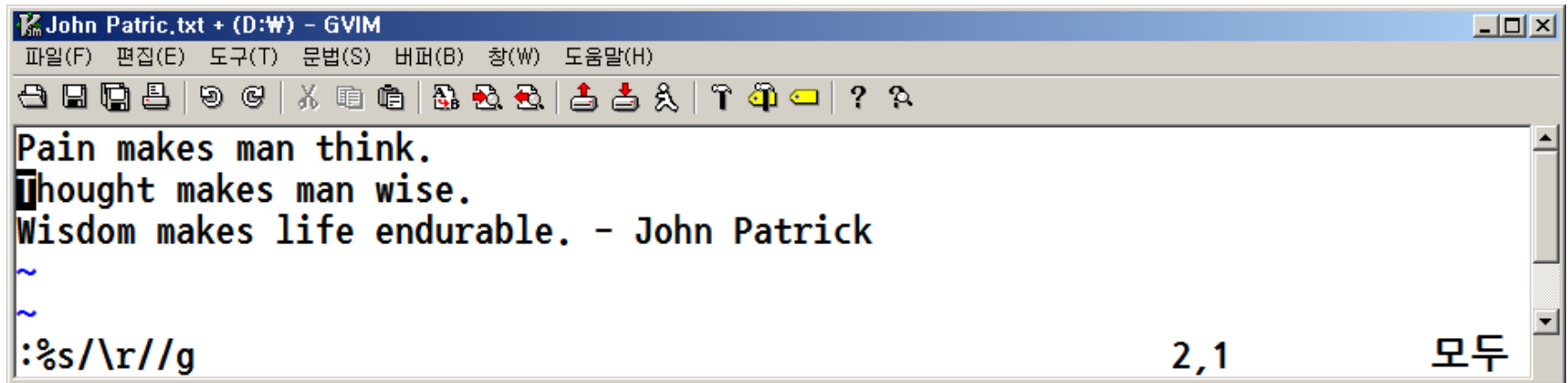
↳ CTRL-V 대신에 CTRL-Q를 사용해도 된다.

substitute : fileformat : DOS/UNIX

❖ CR을 삭제하는 subst 명령

▶ %S/^M//g

▶ %s/\r//g



The screenshot shows the GVIM editor window titled "John Patric.txt + (D:W) - GVIM". The menu bar includes "파일(F)", "편집(E)", "도구(T)", "문법(S)", "버퍼(B)", "창(W)", and "도움말(H)". The toolbar contains various icons for file operations and editing. The text area contains the following text:

```
Pain makes man think.  
Thought makes man wise.  
Wisdom makes life endurable. - John Patrick  
~  
~
```

At the bottom of the window, the command line shows the command `:%s/\r//g` being entered. The status bar on the right indicates the current position as `2,1` and the mode as `모두` (Normal).

Special Char : ASCII

❖ ASCII를 코드값으로 입력하는 경우

입력 방법	설명
<CTRL-V> ###	###에 10진수를 사용하여 ASCII 입력
<CTRL-V> o###	###에 8진수를 사용하여 ASCII 입력
<CTRL-V> x##	##에 16진수를 사용하여 ASCII 입력

Tip!

❖ 숫자 증감

- ▶ 숫자위에서 **CTRL-A**, **CTRL-X**를 누르면 숫자를 즉시 증감시킬 수 있다. 숫자 관련 편집을 할 때 유용한 기능이다.

이 기능은 진수법도 인식하므로 **0x1c** (16진수), **062** (8진수)도 인식.

↳ **CTRL-A** : 1증가

↳ **CTRL-X** : 1감소

substitute : REGEX

❖ 정규표현식으로 교체

- ▶ 행의 맨 끝에
을 추가하려면?

Enabling High Performance Data Transfers

This material is currently under active revision.
Please send any suggestions, additions or corrections to us
at nettune@psc.edu so we can keep the information here as
up-to-date as possible.

URL : <http://www.psc.edu/networking/projects/tcptune/>

~

8,53

All

: %s/\$/
/g

substitute : REGEX (con't)

❖ 정규표현식으로 교체

- ▶ email주소를 HTML 앵커 태그로 교체하는 방법: 먼저 REGEX부터 만들자!

```
Enabling High Performance Data Transfers <BR>
<BR>
This material is currently under active revision. <BR>
Please send any suggestions, additions or corrections to us <BR>
at nettune@psc.edu so we can keep the information here as <BR>
up-to-date as possible. <BR>
<BR>
URL : http://www.psc.edu/networking/projects/tcptune/ <BR>
~
/[a-zA-Z0-9.\-]+\@[a-zA-Z0-9.\-]+\
```

1: 성공	<code>[a-zA-Z0-9.\-]+\@</code>
2: 성공	<code>[a-zA-Z0-9.\-]+\@[a-zA-Z0-9.\-]+\</code>
3: 실패	<code>([a-zA-Z0-9.\-]+\@[a-zA-Z0-9.\-]+\)</code>
4: 성공	<code>\([a-zA-Z0-9.\-]+\@[a-zA-Z0-9.\-]+\)</code>

substitute : REGEX (con't)

❖ 정규표현식으로 교체

- ▶ Back-reference를 사용하기 위해 ()를 사용했는데... = 실패!!

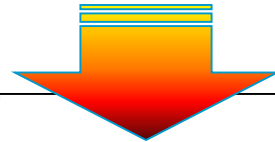
```
<BR>
This material is currently under active revision. <BR>
Please send any suggestions, additions or corrections to us <BR>
at nettune@psc.edu so we can keep the information here as <BR>
up-to-date as possible. <BR>
<BR>
URL : http://www.psc.edu/networking/projects/tcptune/ <BR>
~
E486: Pattern not found: ([a-zA-Z0-9.\-]\+@[a-zA-Z0-9.\-]\+)
Press ENTER or type command to continue
```

() 소괄호는 vim에서 사용하는 meta char.이므로
escape가 필요하다.

substitute : REGEX (con't)

❖ 완성된 REGEX를 바탕으로 Subst. cmd 완성

```
Enabling High Performance Data Transfers <BR>
<BR>
This material is currently under active revision. <BR>
Please send any suggestions, additions or corrections to us <BR>
at nettune@psc.edu so we can keep the information here as <BR>
up-to-date as possible. <BR>
<BR>
URL : http://www.psc.edu/networking/projects/tcptune/ <BR>
~
/\([a-zA-Z0-9.\-]+\@[a-zA-Z0-9.\-]+\)
```



```
up-to-date as possible. <BR>
<BR>
URL : http://www.psc.edu/networking/projects/tcptune/ <BR>
~
:%s/\([a-zA-Z0-9.\-]+\@[a-zA-Z0-9.\-]+\)/<a href="mailto:\1">\1</a>/g
```

Practice 4.1

- ❖ URL을 HTML 앵커 태그로 변환하기 위한 정규표현식 패턴은?

Enabling High Performance Data Transfers

This material is currently under active revision.

Please send any suggestions, additions or corrections to us

at nettune@psc.edu so we can keep the inf
ormation here
up-to-date as

정확한 패턴을 입력하면 이렇게 바뀝니다!

URL : http://www.psc
.edu/networking/projects/tcptune/

~

:%s, , ,g