

# 손에 잡히는 vim (4/4)

김선영

sunyzero@gmail(dot)com

버 전: 2014-11-07

# Ch6. 편리한 편집 기술

쓸만한 것은 이미 다 나왔다. - 괴테

# word forward/backward

## ❖ 단어 경계 이동

```
ret_poll = epoll_wait(epollfd, ep_events, max_open_files, -1);  
if (ret_poll == -1) {  
    /* todo : error */  
}
```

현재 커서 위치

w 키를 누를때마다 멈추는 곳

```
ret_poll = epoll_wait(epollfd, ep_events, max_open_files, -1);  
if (ret_poll == -1) {  
    /* todo : error */  
}
```

현재 커서 위치

e 키를 누를때마다 멈추는 곳

# word forward/backward (con't)

명령어	설명
0	0번째 열
^	공백이 아닌 실제 내용이 있는 시작 열
\$	마지막 열 (행의 끝)
w	단어의 시작 위치 혹은 문장부호의 경계를 따라서 이동 (words forward)
e	w와 같으나, 단어의 끝 부분에 위치합니다. (end of word)
b	w와 비슷하나 진행 방향이 역방향입니다. (words backward)
w, E, B	w, e, b와 비슷하지만 단어가 가진 의미를 따져서 이동합니다.

\* ^와 \$는 정규표현식을 생각해보면 좀 더 쉽게 이해할 수 있다.

# block, paragraph

## ❖ 괄호, 문단, 블록 단위 이동

%

가장 가까운 괄호 짹으로 이동

(프로그래밍시 유용한 기능)

(, )

문장 단위의 시작 위치, 끝 위치로 이동

{, }

문단 단위의 시작 위치, 끝 위치로 이동

[[, ]]

블록 단위의 시작 위치, 끝 위치로 이동

# abbreviation

- ❖ 특정 단어 입력시 대체 입력하는 기능

- ▶ 입력모드에서...

- ↳ "내멜"을 입력하면 sunyzero@gmail.com이 입력되도록,
- ↳ "prjsrc"를 입력하면 /export/2/local/pjR2/mds/1.294/src가 입력되도록,
- ↳ "시간0"을 입력하면 현재 날짜와 시간이 입력되도록,
- ↳ "시간1"이라고 입력하면 로케일(locale)에 맞는 시간이 입력되도록 하고 싶다면

# abbreviation (con't)

- ❖ ab, ia 기능을 이용
  - ▶ ia는 insert mode에서만 작동하는 기능
  - ▶ ca는 commandline mode에서만 작동하는 기능

```
ab 내멜 sunyzero@gmail.com
ab Prjsrc /export/2/local/pjR2/mds/1.294/src
ia 시간0 <C-R>=strftime ("%Y. %m. %d-%H:%M:%S")<CR>
ia tlrks0 <C-R>=strftime ("%Y. %m. %d-%H:%M:%S")<CR>
ia 시간1 <C-R>=strftime ("%c")<CR>
ia tlrks1 <C-R>=strftime ("%c")<CR>
```

# abbreviation (con't)

- ❖ `ca` 기능을 이용하면 한글상태 오타를 변환할 수 있다.

<code>ca z</code>	<code>w</code>
<code>ca zw</code>	<code>wq</code>
<code>ca w</code>	<code>q</code>
<code>ca =</code>	<code>x</code>

- ▶ 한글 약어 매크로 사용시 주의점

- ↳ `.vimrc` 파일의 인코딩 형식에 주의!!

- ▶ 간혹 `.vimrc` 파일이 EUC-KR로 되어있으면 UTF-8 문서 편집시 작동하지 않을 수 있다.

# abbreviation : cmd

## 명령어

## 설명

**:ab [lhs]**

현재 설정된 모든 약어 목록을 출력합니다.

lhs에 약어를 지정하면 해당 약어의 정보만 출력합니다.

**:ab {lhs} {rhs}**

약어 lhs를 rhs로 변환합니다.

**:unab {lhs}**

약어 lhs를 해제합니다.

**:abclear**

설정된 모든 약어를 해제합니다.

**:ia {lhs} {rhs}**

ab와 기능은 같지만 입력 모드에서만 작동합니다.

**:ca {lhs} {rhs}**

ab와 기능은 같지만 명령행 모드에서만 작동합니다.

# register

register = vim temp. storage

# vim register

## ❖ vim에서 사용하는 임시 버퍼

```
:reg
--- Registers ---
"" Wisdom makes life endurable
"0 PREFIX_DIR^I= ../../..
"1 OBJS      = $(patsubst %.c,%.o,$(SRCS))^J
"2 PREFIX_DIR^I= ../../..
"3 /* vim: set ts=4 sw=4: */^J
"4 ^Isaddr_in.sin_port = htons(((short)atoi(argv[2])));^J
"5 for (i=0; i<LOOP_ITERATION; i++) {^Jx = (double)rand() * rns;^Jy = (doub
"6 up-to-date as possible.^J
"7 mymailis^Isunyzero@gmail.com^Isunyzero@gmail.com ^J
"8 work/pi.c^J
"9 pi.c^J
"- Wisdom makes life endurable
": "
":%"
"/ " 부터 / 까지 다양한 레지스터가 등록되어있습니다
\\(http://[a-zA-Z0-9._%&=-?/-]\+\)
Press ENTER or type command to continue
```

# vim register : list

레지스터 이름	설명
""	가장 최근에 복사, 삭제된 데이터
"0	가장 최근에 복사(yank)된 데이터
"1 ~ "9	가장 최근에 삭제된 데이터(시간순, 1번이 가장 최근 데이터)
a~z	사용자 지정 레지스터

# vim register : list (con't)

레지스터 이름	설명
"-	가장 최근에 한 라인 이내로 삭제한 데이터
"/	가장 최근에 검색한 데이터
가장 최근에 명령행 모드에서 내린 명령어 데이터 (읽기 전용)	
가장 최근에 입력한 데이터 (읽기 전용)	

레지스터 이름	설명
"%	현재 편집 중인 파일명 (읽기 전용)
"#	이전에 열었던 파일명. <CTRL-^>을 누르면 열게 될 파일. (읽기 전용)

# register : cmd

## ❖ 레지스터 명령어 요약

명령어	설명
" {reg} y{motion}	{reg} 레지스터에 {motion}에 해당하는 부분을 복사합니다.
" {reg} p	{reg} 레지스터의 내용을 현재 커서 뒷부분에 넣습니다. (레지스터에 개행 문자가 포함된 경우는 커서의 아래 행에 넣습니다.)
" {reg} P	{reg} 레지스터의 내용을 현재 커서의 앞부분에 넣습니다. (레지스터에 개행 문자가 포함된 경우는 커서의 윗 행에 넣습니다.)
" {reg} d{motion}	{reg} 레지스터에 {motion}에 해당하는 부분을 잘라 넣습니다. * 참고 : vim에서 삭제하는 모든 행위는 잘라내기로 작동합니다.
CTRL-R{reg}	입력모드에서만 사용하는 명령으로 {reg} 레지스터를 붙여 넣습니다.

\* 참고 : "{reg}"를 생략하면 "" 레지스터를 가리킵니다.



# Tip!

---

❖ vim register는 vim이 종료하면?

- ▶ vim register는 .viminfo 파일에 저장해두므로, 다음번 실행할 때도 불러오게 된다.
- ↳ 윈도용은 \_viminfo 파일에 기록된다.

# Ch7. 자동화

반복은 연구의 어머니다 - 쇼펜하우어, 문장론



# key map

---

- ❖ nmap <단축키> <명령>
  - ▶ normal mode에서만 작동
- ❖ imap <단축키> <명령>
  - ▶ insert mode에서만 작동
- ❖ map <단축키> <명령>
  - ▶ e.g. nmap <F2> :up<CR>
  - ▶ e.g. imap <F4> sunyzero@gmail.com

# key map : list

## 명령어

## 설명

**nmap key command** 일반 모드에서 key를 누르면 command를 실행합니다.

**imap key command** 입력 모드에서 key를 누르면 command를 실행합니다.

**vmap key command** 비주얼 모드에서 key를 누르면 command를 실행합니다.

**cmap key command** 명령행 모드에서 key를 누르면 command를 실행합니다.



# map : example

---

- ❖ .vimrc 파일

```
"[ map ]  
nmap <F2>      :up<CR>  
nmap <F3>      :tabnext<Enter>  
nmap <F4>      :tabprev<Enter>
```

# autocmd

❖ 특정 상황에서 자동으로 실행할 명령

▶ 예를 들어 \*.c 파일을 열 때 자동으로 실행되어야 하는 명령

```
autocmd BufRead,BufNewFile *.txt colo evening  
autocmd BufRead,BufNewFile *.java colo morning|set ts=2 sw=2
```

```
au SwapExists * let v:swapchoice = 'o'
```

- ↳ v:swapchoice 스왑 파일이 존재하는 경우의 선택 (o : readonly)
- ↳ v:swapchoice에는 e, q 등등을 넣을 수 있다.

# vimrc 의 예시 #2

## ❖ .vimrc 예시의 발전된 형태

```
" example : vim runtime configuration  
set ai cindent  
set ts=4 sw=4  
" set nu  
set fencs=ucs-bom,utf-8,korea  
colorscheme desert  
nmap <F2> :w!<ENTER>  
nmap <F3> :e .<CR>
```

현재 디렉토리를 열어준다.  
즉 파일 선택창이 제공된다.

# Recording

q (recording)

@ (execute)



# Recording

❖ 매크로 작업 : 반복 작업을 녹화 후 실행

▶ 매크로 이름은 {register}를 의미한다.

▶ q{register}

↳ vim register에 작업을 녹화

▶ @{register}

↳ vim register에 기록된 작업을 실행

↳ @@는 이전 실행을 repeat

▶ register는 a-z를 사용한다.

# Practice

- ❖ clientlist.txt와 clientmail.txt를 준비

1304, Yona Yahav, M, 42, MP1

1294, Kebin Robinson, M, 41, CP1

1601, Steven Choi, M, 34, CP3

1314, TW Yoon, F, 46, CP1

1315, Rina Suzuki, F, 36, MP1

1600, Robert Kim, M, 32, CP3

1297, Rarry Robinson, M, 38, CP2

1600, optiger22@ibn.com

1314, teawooy1@boggle.com

1315, rinzzang@yahoo.co.jp

1601, schoi74@idontcare.net

1304, yoyohave@vimvim.co.kr

1294, braverobb@bettervi.com

1297, rrjr4624@orakle.com

# Practice (con't)

## ❖ 분할 창으로 열기

▶ vim -o clientlist.txt clientmail.txt

```
1304, Yona Yahav, M, 42, MP1  
1294, Kebin Robinson, M, 41, CP1  
1601, Steven Choi, M, 34, CP3  
1314, TW Yoon, F, 46, CP1  
1315, Rina Suzuki, F, 36, MP1  
1600, Robert Kim, M, 32, CP3
```

clientlist.txt

1,1

Top

```
1314, teawooy1@boggle.com  
1315, rinzzang@yahoo.co.jp  
1601, schoi74@idontcare.net  
1304, yoyohave@vimvim.co.kr  
1294, braverobb@bettervi.com  
1297, rrjr4624@orakle.com
```

clientmail.txt

7,1

Bot

# Practice (con't)

## ❖ 녹화 시작 : a 레지스터에 녹화

```
1304, Yona Yahav, M, 42, MP1  
1294, Kebin Robinson, M, 41, CP1  
1601, Steven Choi, M, 34, CP3  
1314, TW Yoon, F, 46, CP1  
1315, Rina Suzuki, F, 36, MP1  
1600, Robert Kim, M, 32, CP3
```

『qa』를 눌러서 “레지스터 a”에 녹화 시작

```
clientlist.txt  
1314, teawooy1@boggle.com  
1315, rinzzang@yohoo.co.jp  
1601 schoi74@idontcare.net  
13 『qa』를 누르면 co.kr  
12 recording 이 표시됨 vi.com  
11 ... .com  
clientmai...txt  
recording
```

1,1 Top

7,1 Bot

# Practice (con't)

## ❖ 고객번호 복사

```
1304, Yona Yahav, M, 42, MP1  
1204, Kehin Robinson, M, 41, CP1  
1 『yiw』 Jen Choi, M, 34, CP3  
1314, Iw Yoon, F, 46, CP1  
1315, Rina Suzuki, F, 36, MP1  
1600, Robert Kim, M, 32, CP3
```

clientlist.txt

```
1314, teawooy1@boggle.com  
1315, rinzzang@yahoo.co.jp  
1601, schoi74@idontcare.net  
1304, yoyohave@vimvim.co.kr  
1294, braverobb@bettervi.com  
1297, rrjr4624@orakle.com
```

clientmail.txt

recording

1. 『^』를 눌러서 행의 맨 앞으로 이동한다.
2. 행의 맨 앞에서 『yiw』를 눌러 번호를 복사한다.

Top

5,1

Bot

# Practice (con't)

## ❖ clientmail.txt에서 고객번호 검색

```
1304, Yona Yahav, M, 42, MP1  
1294, Kebin Robinson, M, 41, CP1  
1601, Steven Choi, M, 34, CP3  
1314, TW Yoon, F, 46, CP1  
1315, Rina Suzuki, F, 36, MP1  
1600, Robert Kim, M, 32, CP3
```

clientlist.txt

1314, teawooy1@boggle.com

『/〈CTRL-R〉』를  
누르면 /1304가  
표시됩니다.

1291, r...+624@orakle.com

clientmail.txt

/1304

- 『〈CTRL-W〉 w』로 clientmail.txt로 이동
- 『/』을 눌러서 검색 기능 호출
- 『〈CTRL-R〉 “』으로 이전에 복사한 번호 붙여넣기
- 『〈Enter〉』로 검색 시작

Top

7,1

Bot

# Practice (con't)

## ❖ 메일 주소 부분 복사

1304, Yona Yahav, M, 42, MP1

1294, Kebin Robinson, M, 41, CP1

1601, Steven Choi, M, 34, CP3

1314, TW Yoon, F, 46, CP1

1315, Dina Suzuki, F, 36, MP1

『2w』를 누르면

메일 주소에 커서가

위치합니다.

1. 『2w』로 메일 주소 시작 부분에 커서 위치

2. 『y\$』을 눌러서 행의 끝까지 복사

Top

1515, 12345@angryahoo.co.jp

1601, schoi74@idontcare.net

1304, yoyohave@vimvim.co.kr

1294, braverobb@bettervi.com

1297, rrjr4624@orakle.com

clientmail.txt

recording

5,7

Bot

# Practice (con't)

## ❖ clientlist.txt에 복사한 메일 주소 붙여넣기

1304, Yona Yahav, M, 42, MP1, yoyohave@vimvim.co.kr

1294, Kebin Robinson, M, 41, CP1

1601, Steven Choi, M, 34, CP3

1314, TW Yoon, F, 46, CP1

1315, Rina Suzuki, F, 36, MP1

1600, Robert Kim, M, 32, CP3

clientlist.txt [+]

1314, teawooy1@boggle.com

1315, rinzzang@yahoo.co.jp

1601, schoi74@idontcare.net

1304, yoyohave@vimvim.co.kr

1294, braverobb@bettervi.com

1297, rrjr4624@orakle.com

clientmail.txt

recording

- 『<CTRL-W> w』로 clientlist.txt로 돌아감
- 『A』을 행의 끝으로 이동하면서 입력 모드로 전환
- 콤마를 입력 후 『<ESC>』를 눌러 입력 모드 종료
- 『p』를 눌러 메일 주소를 붙여 넣음
- 『q』를 눌러 녹화 종료

Top

5,7

Bot

# Practice (con't)

## ❖ 녹화된 매크로 재생

```
1304, Yona Yahav, M, 42, MP1, yoyohave@vimvim.co.kr  
1294, Kebin Robinson, M, 41, CP1, braverobb@bettervi.com  
1601, Steven Choi, M, 24, CP3  
1314, TW 2번째 행에서 「@a」를  
1315, Ri 누르면 a에 녹화된  
1600, Ro 내용이 재생됩니다.
```

clientlist.txt [+]

2,56

Top

```
1314, teawooy1@boggle.com  
1315, rinzzang@yahoo.co.jp  
1601, schoi74@idontcare.net  
1304, yoyohave@vimvim.co.kr  
1294, braverobb@bettervi.com  
1297, rrjr4624@orakle.com
```

clientmail.txt

6,7

Bot

순서	명령어	설명
0.	<b>qa</b>	a 레지스터에 녹화를 시작합니다.
1.	^	행의 맨 앞으로 이동합니다.
2.	<b>yiw</b>	숫자 부분을 복사합니다.
3.	<b>CTRL-W CTRL-W</b>	clientemail.txt 파일로 이동합니다.
4.	/	검색 기능을 사용합니다.
5.	<b>CTRL-R "</b>	최근에 복사한 레지스터를 검색어로 사용합니다.
6.	<b>&lt;ENTER&gt;</b>	검색을 시작합니다.
7.	<b>2w</b>	문장 부호를 2개를 뛰어 넘어 메일 주소 부분으로 이동합니다.
8.	<b>y\$</b>	메일 주소를 복사합니다.
9.	<b>CTRL-W CTRL-W</b>	clientlist.txt 파일로 돌아갑니다.
10.	<b>A</b>	행의 맨 끝으로 이동하면서 입력 모드로 변경합니다.
11.	,	쉼표를 입력합니다.
12.	<b>&lt;ESC&gt;</b>	입력 모드를 종료합니다(레지스터에는 ^[로 표시됩니다).
13.	<b>p</b>	앞에서 복사한 메일 주소를 붙여넣습니다.
14.	<b>q</b>	녹화를 종료합니다.

# Practice #2

# ❖ **기획된 내용 확인**

1314, TW Yoon, F, 46, CP1  
1315, Rina Suzuki, F, 36, MP1  
1600, Robert Kim, M, 32, CP3

clientlist.txt +

1,1

Top

1314, teawooy1@boggle.com

1315, rinzzang@yahoo.co.jp

1601 schoi74@idontcare.net

130 a 레지스터의 내용을 co.kr

129 확인해 봅니다. vi.com

129, 1117 [www.karate.com](http://www.karate.com)

clientmail.txt

6,7

Bot

:reg a

--- Registers ---  
"a ^yiw^W^W/^R" ^M2wy\$^W^WA, ^[p

Press ENTER or type command to continue



# Practice #2 (fin)

---

❖ clientlist.txt 파일을 열고:

▶ :%normal! ^^X

↳ ^^X : ^ + <CTRL-V><CTRL-X>

▶ 어떤 일이 발생하는가?

↳ 이번 5씩 감소시키려면?



## Practice #3

---

- ❖ 녹화된 내용의 수정
  - ▶ 다시 녹화를 하는 것은 비효율적이다.
- ❖ 예를 들어 **clientlist.txt**의 매크로에서 메일 주소 부분을 "..."으로 감싸달라고 한다. 어떻게 해야 하는가?
  - ▶ 변환 전 상태에서...

# Practice #3 (con't)

- ❖ 녹화된 내용을 불러온다.

```
1601, Steven Choi, M, 34, CP3
```

```
1314, TW Yoon, F, 46, C
```

```
1315, Rina Suzuki, F, 32, C
```

```
1600, Robert Kim, M, 32, C
```

```
1297, Rarry Robinson, M, 38, C
```

```
^yiw^W^W/^R"^-M2wy$^W^WA,
```

```
clientlist.txt [+]
```

```
1314, teawooy1@boggle.com
```

```
1315, rinzzang@yahoo.co.jp
```

```
1601, schoi74@idontcare.net
```

```
1304, yoyohave@vimvim.co.kr
```

```
1294, braverobb@bettervi.com
```

```
1297, rrjr4624@orakle.com
```

```
clientmail.txt
```

빈 행을 만들고  
레지스터 a를 붙입니다.

^[p]

^[ : <ESC>  
p : put (붙여넣기)

변경 전 : ^[p  
변경 후 : "^[pa"^[

Bot

6,7

Bot

수정 후 행의 맨 앞에서 "ay\$

# Practice #3 (con't)

- ❖ 녹화된 내용을 수정한다.

```
1314, TW Yoon, F, 46, CP1  
1315, Rina Suzuki, F, 36, MP1  
1600, Robert Kim, M, 32, CP3
```

```
clientlist.txt [+]
```

1,1

Top

```
1314, teawooy1@boggle.com  
1315, rinzzang@yahoo.co.jp  
1601, schoi74@idontcare.net  
1304, yoyohave@vimvim.co.kr  
1294, braverobb@bettervi.com  
1297, rrjr4624@orakle.com
```

```
clientmail.txt
```

수정된 부분

7,7

Bot

```
:reg a
```

```
--- Registers ---
```

```
"a ^yiw^W^W/^R" ^M2wy$^W^WA, "^[pa"^[
```

```
Press ENTER or type command to continue
```



# Tip!

---

- ❖ 재생할 작업이 1,000개라면?
  - ▶ 수동으로 @a나 @@을 천번 눌러야 하나?
- ❖ 두가지 방법이 존재한다.
  - ▶ 반복수 지정
    - ↳ 1000@a
  - ▶ 일반모드의 명령을 수행하는 :normal 명령을 수행
    - ↳ :2,\$normal! @a

# Recording : cmd

## ❖ 레코딩 관련 명령어 정리

명령어	설명
q{매크로 이름}	녹화를 시작합니다. 매크로 이름은 {a-zA-Z} 중 선택해야 합니다.
@{매크로 이름}	매크로를 실행합니다. 매크로 이름은 {a-z} 중 선택해야 합니다.
@@	바로 이전에 실행된 매크로를 재실행합니다.
: {range} <b>norm!</b> @{매크로 이름} {range}에 해당하는 범위에 매크로를 실행한다.	

# Practice #4

- ▶ 현재 디렉터리에 있는 모든 \*.txt 파일에서 ‘유닉스’라는 단어를 찾아서 ‘리눅스’로 바꾸는 기능을 녹화 기능을 이용해서 완성해 봅시다.

순서	명령어	설명
1.	:n *.txt	현재 디렉터리의 *.txt 파일을 파일 목록에 등록합니다.
2.	qb	b 레지스터에 녹화합니다.
3.	빈칸	유닉스를 리눅스로 바꿉니다. (에러 발생시 무시합니다.)
4.	빈칸	저장합니다.
5.	빈칸	다음 파일로 이동합니다.
6.	q	녹화를 종료합니다.
7.	999@b	b 레지스터를 999번 재생합니다. (현재 디렉터리의 *.txt 파일이 999개 미만이라고 가정합니다.)

# Ch8. 프로그래머에게 유용 한 기능



추가 요금 없이 난 자네에게 자연의 역사에 관한 많은  
부분을 가르쳐 주겠네 - 루이스 캐럴

# re-indenting

## ❖ 들여쓰기를 다시 하기

```
t_now = time(0);
srand((unsigned int)t_now);
for (i=0; i<LOOP_ITERATION; i++) {
    x = (double)rand() * rns;
    y = (double)rand() * rns;
    if (x*x + y*y < 1) {
        hits++;
    }
}
printf("pi = %f\n",
```

들여쓰기가 안되어있거나  
들여쓰기가 밀린 경우...  
수동으로 한땀한땀 한다면 얼마나 걸릴까?

```
    if (timer_create(CLOCK_REALTIME, &sigev, &rt_timer) == -1) {
        perror("FAIL: timer_create()");
        return -1;
    }
    rt_itspec.it_value.tv_sec = 2; /* timer expiration after 2.5 sec. */
    rt_itspec.it_value.tv_nsec = 5000000000; /* 0.5 sec */
    rt_itspec.it_interval.tv_sec = 4; /* periodic timer with 4 se
c. */
    rt_itspec.it_interval.tv_nsec = 0;
    printf("Enable timer at %s.\n", GET_TIME0(ts_now));
    if (timer_settime(rt_timer, 0, &rt_itspec, NULL)
= -1) {
```

# re-indenting (con't)

- ❖ ={motion}
  - ▶ motion에는 이동 관련 키 gg, G, ), ], ]] 등을 사용할 수 있다.
- ❖ visualmode + =
  - ▶ 비주얼모드로 라인 선택 후 = 키를 눌러도 된다.

# tab vs whitespace

- ❖ tab 문자의 가변적인 길이 때문에 공백 #칸을 대신 사용
  - ▶ 다양한 플랫폼에서 개발되는 프로젝트의 경우
  - ▶ 일반적으로 tab 대신에 공백 4칸을 사용하는 경우 :

```
" .vimrc (vim runtime config.)  
set et ts=4
```

- ↳ et : expandtab, 탭 대신 공백 사용
- ↳ ts : tabstop, 탭 크기

expandtab 환경에서 진짜 탭 문자를 입력하려면  
**<CTRL-V><TAB>** 혹은 **<CTRL-V><CTRL-I>**를 사용한다.

# tab vs whitespace (con't)

- ▶ 기존에 작성된 탭 문자를 공백 4칸으로 전환하고 싶다면?

```
:set et ts=4  
:ret
```

↳ ret (retab) 대신에 re-indenting 기능을 사용하는 편이 좋다.

- ▶ 예제 프로그램

```
int main(int argc, char *argv[])
{
    if (argc == 1) {
        printf("argc == 1\n");
    }
    return 0;
}
```

# tab vs whitespace (con't)

- ▶ 반대로 공백 4칸을 탭 문자로 변환하려면?

```
:set noet ts=4  
:ret!
```

# insert-completion

## ❖ 자동 완성 기능

```
/* hellomonkey.c by sunyzero@gmail.com */

#include <stdio.h>
#include <stdlib.h>
int main()
{
    printf("Hello. Mr. monkey\n");
    return 0;
}
```

mo까지만 입력한 뒤에  
<CTRL-N>을 누르면?

# insert-completion (con't)

- ❖ vim은 #include 구문도 이해한다.

- ▶ <CTRL-N> : 단어완성을 위해서 현재 문서와 관련 파일을 전방탐색
- ▶ <CTRL-P> : <CTRL-N>과 반대방향으로 탐색

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    printf("Hello world\n");
    exit(EXIT_FAILURE)
}
```

<CTRL-N>으로

단어완성을 한 결과 = **stdlib.h**를 탐색한다.

자동 완성 탐색 결과로

"**2개** 中 첫 번째" 임을 보여줌

```
-- Keyword completion (^N^P) match 1 of 2
```

# insert-completion (con't)

- ❖ 자동 완성 기능 (특수 문자가 포함된 경우라면?)

```
/* hellomonkey.c by sunyzero@gmail.com */  
#include <stdio.h>  
#include <stdlib.h>  
int main()  
{  
    printf("Hello. Mr. monkey\n");  
    return 0;  
}
```

sunyzero@gmail.com을 검색하려면?  
**suny<CTRL-N><CTRL-X><CTRL-N> ...**

# insert-completion (con't)

## ❖ 사전 기반 검색, 파일명 검색

<CTRL-X> <CTRL-N>

더하기 낱말 모드로 작동하여 추가 검색을 합니다.

원하는 낱말이 아닌 경우에는 <CTRL-N>을 계속 누르면 됩니다.

<CTRL-X> <CTRL-K> <CTRL-N>

사전 검색 모드로 작동합니다.

▶ 파일명을 몇 글자 타이핑 후에 <CTRL-X><CTRL-F>을 누르면?

↳ 파일명 검색을 할 수 있다.(파일명 타이핑시 오타 방지)

```
~/.bash_history  
~/.bash_history  
~/.bash_logout  
~/.bash_profile  
~/.bashrc  
~
```

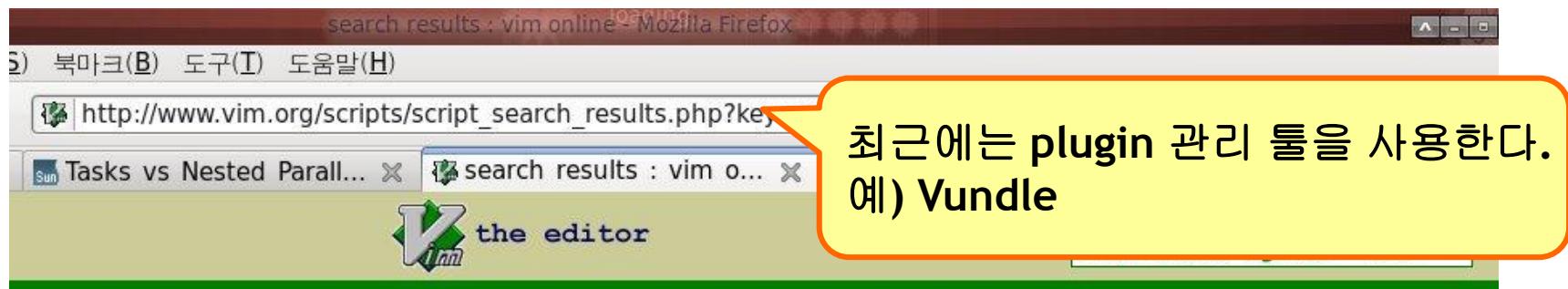
~/ba 까지만 입력한  
상태에서 파일명 검색

# Ch9. 플러그인

태양 아래 새로운 것은 없나니 - 솔로몬

# vim plugin

- ❖ vim의 플러그인은 ~/.vim 디렉터리에 넣어두면 되는데...



최근에는 plugin 관리 툴을 사용한다.  
예) Vundle

search results : vim online Mozilla Firefox

5) 북마크(B) 도구(I) 도움말(H)

http://www.vim.org/scripts/script\_search\_results.php?key=

Tasks vs Nested Parall... search results : vim o...  the editor

Search Results

Searched scripts for ""

Vundle 설치법  
<http://sunyzero.tistory.com/212>

Script	Type	Downloads	Comments	Description
taglist.vim	utility	7405	158313	<a href="#">php, etc</a>
The NERD tree	utility	3968	61830	<a href="#">A tree explorer plugin for navigating the filesystem</a>
bufexplorer.zip	utility	2498	61819	<a href="#">Buffer Explorer / Browser</a>
minibufexpl.vim	utility	3207	58037	<a href="#">Elegant buffer explorer - takes very little screen space</a>
c.vim	utility	3419	52986	<a href="#">C/C++ IDE -- Write and run programs. Insert statements, idioms, comments etc.</a>
project.tar.gz	utility	2776	50084	<a href="#">Organize/Navigate projects of files (like IDE/buffer explorer)</a>
Color Sampler Pack	color scheme	2349	50039	<a href="#">*NEW* [Jan 2010 Update] Top 100 Themes, GUI Menu</a>



# 유용한 플러그인

---

- ❖ Vundle : <http://sunyzero.tistory.com/212>

- ▶ 플러그인을 관리하는 플러그인

- ❖ Airline

- ▶ 상태바와 화면을 예쁘게 꾸며주는 플러그인

- ↳ 예전에 사용되던 powerline의 업그레이드 버전

- ❖ AutoComplPop

- ▶ 자동완성창을 띄워주는 플러그인 : 프로그래밍시 유용

# 단축키 #1

<http://sunyzero.tistory.com/131>

**Vim 명령어 단축키**

여기 있는 Vim 명령어

**Esc**  
명령 모드

~ 대소문자 전환! 오부 명령@ 매크로 실행# 이전 검색\$ 행 끝으로 이동% 맵 글호 찾기^ 첫 글자& 번역★ 다음 검색( 문장 시작) 문장 끝- 아래 행+ 다음 행  
‘ 표식으로 이동12345678900 앞으로= 이전 행  
Q 실행 모드W 다음 단어 이동E 단어 이동R 수정 모드T 행에서 후방 검색Y 행 단위 복사U 행 단위 헤스I 행 시작에 삽입O 행 위에 삽입P 커서 이전에 불어넣기{ 문단 시작} 문단 끝  
q 매크로 기록w 단어 단어 이동e 단어 글자 이동r 한 문자 교체t 행에서 전방 검색y 복사u 실행 취소i 커서 앞에 삽입o 행 아래에 삽입p 커서 이후에 불어넣기[ 문단 시작] 문단 끝  
A 행 끝에 삽입S 삭제 후 입력 모드D 삭제F 행에서 후방 검색G 파일로 이동H 화면 상단J 아래 행 합침K 도움말L 화면 하단: 명령행 모드;" 레지스터 지정I 열 이동  
a 커서 뒤에 삽입s 단어 삭제 후 입력 모드d 삭제f 행에서 전방 검색g 확장 영역h ←j ↓k ↑l →t/T/f/F 번역' 표식으로 이동\ 사용하지 않음

**동작** 커서를 이동하거나,  
연산자가 동작할 범위를 지정합니다.

**명령** 바로 동작하는 명령입니다.  
**빨간색**은 입력 모드로 변경됩니다.

**오퍼레이션**  
**캔슬 모드** 커서 위치부터 목적지까지를 대상으로  
명령을 실행합니다.

**화장** 추가적인 키 입력이 필요합니다.

**q-** 입력 후 글자를 입력해야 합니다.

**단어** 공백 문자나 특수 문자로 구분된 단어  
`test(123, 456, 789);`

**단어 (파라미)** 공백 문자로 구분된 단어  
`test(123, 456, 789);`

- 명령행 모드의 주요 명령어
- `:w` 저장
- `:q` 종료
- `:q!` 차등없이 종료
- `:ef` 파일 열기
- `:%s/x/y/g` 파일 전체에서 x 를 y 로 교체
- `:h name` 명령에 대한 도움말
- `:new` 새 파일

- 일반 모드의 주요 명령어
- `CTRL-R` 재실행
- `CTRL-F/-B` 페이지 위로/아래로
- `CTRL-E/-Y` 스크롤 위로/아래로
- `CTRL-V` 비주얼 모드

- 참고
- ❶ **복사/붙여넣기/자우기** 명령을 사용하기 전에  
“a”를 입력하여 레지스터 a를 지정할 수 있습니다.  
(레지스터 이름은 a부터 z까지 사용 가능)  
예를 들어 “ay\$”는 커서 위치부터 행 끝까지의 내용을  
레지스터 a에 저장합니다.
- ❷ 명령어 입력 전 숫자를 지정하면,  
해당 숫자만큼 명령어가 반복됩니다.
- ❸ 연속으로 입력하면, 현재 행에 반영됩니다.  
예를 들어 dd는 현재 행이 지워집니다.
- ❹ ZZ는 저장 후 종료, ZQ는 저장 없이 종료입니다.
- ❺ zi는 커서가 위치한 부분을 화면 상단으로 스크롤합니다.  
zb는 바닥으로, zz는 가운데로 스크롤합니다.
- ❻ gg는 커서를 파일 처음으로 이동합니다.  
gi는 커서 위치의 파일명을 인식하여 파일을 엽니다.

• 출처 : [www.viemu.com](http://www.viemu.com) • 번역 : Mr. Dust

# 단축키 #2

<http://sunyzero.tistory.com/131>

## Vim 이동 단축키

◀ 앞  
0 ▲  
행 처음으로 행 첫 글자

**Fx** 이전 x  
**Tx** 이전 x  
문자로  
문자 뒤로

**b**      **ge**  
이전 단어      이전 단어  
**b**      **ge**  
앞 단어      앞 단어

10

1

e w

tx

10

**CTRL-d** 아래로 한 페이지 스크롤    **CTRL-f** 아래로 한 페이지 스크롤

/text n \*

마지막

손에 잡히는 Vim 안내서

### • 이동키

#G %

\* #21 : <https://bitbucket.org/technoaid/vim-shortcut-wallpaper>

# Practice #4 : solution

- ▶ 현재 디렉터리에 있는 모든 \*.txt 파일에서 ‘유닉스’라는 단어를 찾아서 ‘리눅스’로 바꾸는 기능을 녹화 기능을 이용해서 완성해 봅시다.

순서	명령어	설명
1.	:n *.txt	현재 디렉터리의 *.txt 파일을 파일 목록에 등록합니다.
2.	qb	b 레지스터에 녹화합니다.
3.	:s/UNIX/Linux/ge	유닉스를 리눅스로 바꿉니다. (에러 발생시 무시합니다.)
4.	:w	저장합니다.
5.	:n	다음 파일로 이동합니다.
6.	q	녹화를 종료합니다.
7.	999@b	b 레지스터를 999번 재생합니다. (현재 디렉터리의 *.txt 파일이 999개 미만이라고 가정합니다.)

주의 사항 : 유닉스란 단어가 하나도 등장하지 않는 텍스트 파일이 있다면 에러가 발생하고, 매크로 실행이 즉시 중단된다. 이를 방지하기 위해서 교체 명령어에 에러를 무시하도록 플래그를 추가한 것입니다.