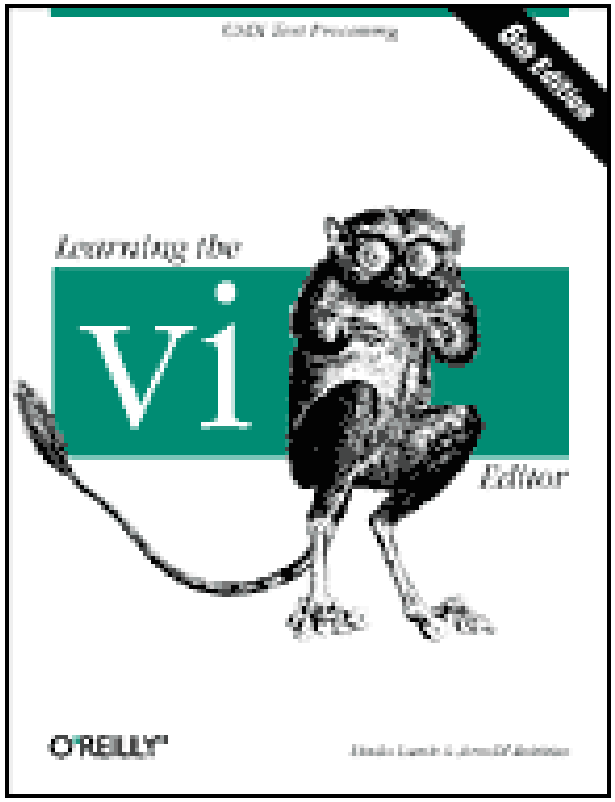
Development Environment 1
vi Editor

Jo, Heeseung

Suggested Reading for vi

If your really need to know about vi,

- Try "Learning the vi Editor" from Oreilly.Com



What is vi?

The visual editor on the Unix

Before vi the primary editor used on Unix was the line editor

- User was able to see/edit only one line of the text at a time

The vi editor is not a text formatter (like MS Word, Word Perfect, etc.)

- You cannot set margins
- Center headings
- Etc. …

vi History

Although other stories exist,
the true one tells that vi was originally
written by Bill Joy in 1976

Who is Bill Joy you ask?

- He co-founded Sun Microsystems in 1982 and served as chief scientist until 2003

Joy's prowess as a computer programmer is legendary, with an anecdote that he wrote the vi editor in a weekend

- Joy denies this assertion

Some vi Ports

All Unix OS's

MS-Dos

Windows 3.x

Windows 9x/2k/NT/XP

OS/2

Blah blah blah ...

Macintosh

Atari

Amiga

OpenVMS/Alpha

OpenVMS/VAX

Characteristics of vi

The vi editor is:

- A very powerful
- But at the same time it is cryptic
- It is hard to learn, specially for windows users

The best way to learn vi commands is to use them

So practice... practice... practice...

vim equals vi

The current iteration of vi for Linux is called vim

- vi improved
- http://www.vim.org

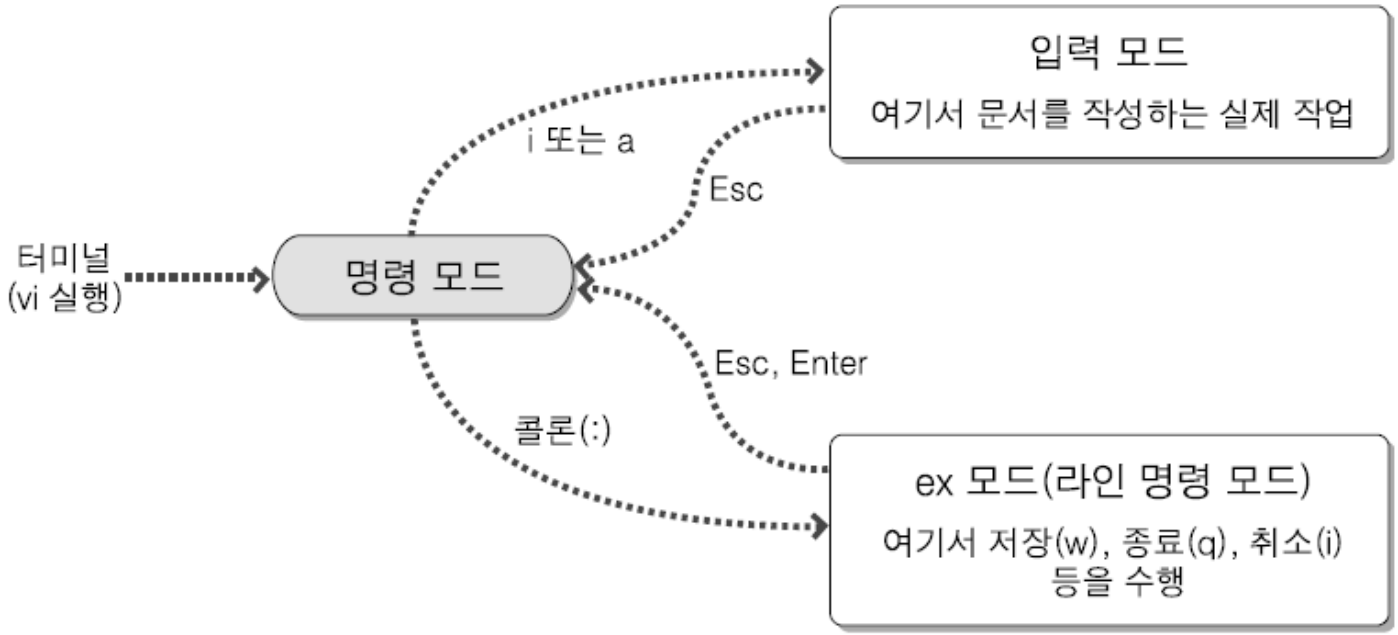
vi edit의 모드

Command mode

- Text의 위치나 편집 가능하게 함

Input mode

- 사용자가 text를 작성할 수 있는 모드



How to exit from vi (command mode)

:q <enter>

- To exit, if you have not made any changes to the file

:q! <enter>

- The forced quit, it will discard the changes and quit

:wq <enter>

- For save and Exit

:x <enter>

- Same as above command

ZZ

- For save and Exit (Note this command is uppercase)

The ! Character forces over writes, etc. :wq!

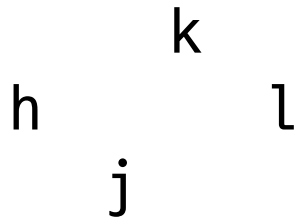
Moving Around

You can move around only when you are in the command mode

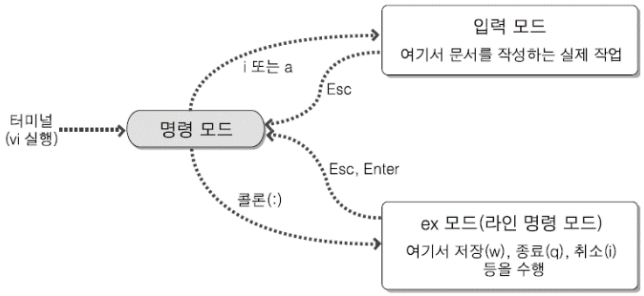
Arrow keys usually works (but may not)

The standard keys for moving cursor are:

- h - for left
- l - for right
- j - for down
- k - for up



Entering text



To enter the text in vi you should first switch to input mode

- To switch to input mode there are several different commands
- a – Append mode places the insertion point after the current character
- i – Insert mode places the insertion point before the current character
- I – places the insertion point at the beginning of current line
- o – is for open mode and places the insertion point after the current line
- O – places the insertion point before the current line
- R – starts the replace(overwrite) mode

Editing text

x - deletes the current character

d - is the delete command but pressing only d will not delete anything you need to press a second key

- dd - deletes the current line
- dw - deletes to end of word
- d0 - deletes to beginning of line

There are many more keys to be used with delete command

Undo and repeat command

u - undo the changes made by editing commands

. - repeats the last edit command

Copy, cut and paste in vi

yy / dd - (yank) copy / cut current line to buffer

nyy / ndd - where n is number of lines

p - paste the yanked lines from buffer to the line below

v - visual mode

y - yank

P - paste the yanked lines from buffer to the line above

(the paste commands will also work after yank)

Jump

gg - jump the first line of file

G - jump the end of file

nG - jump to the line (where n is line number)

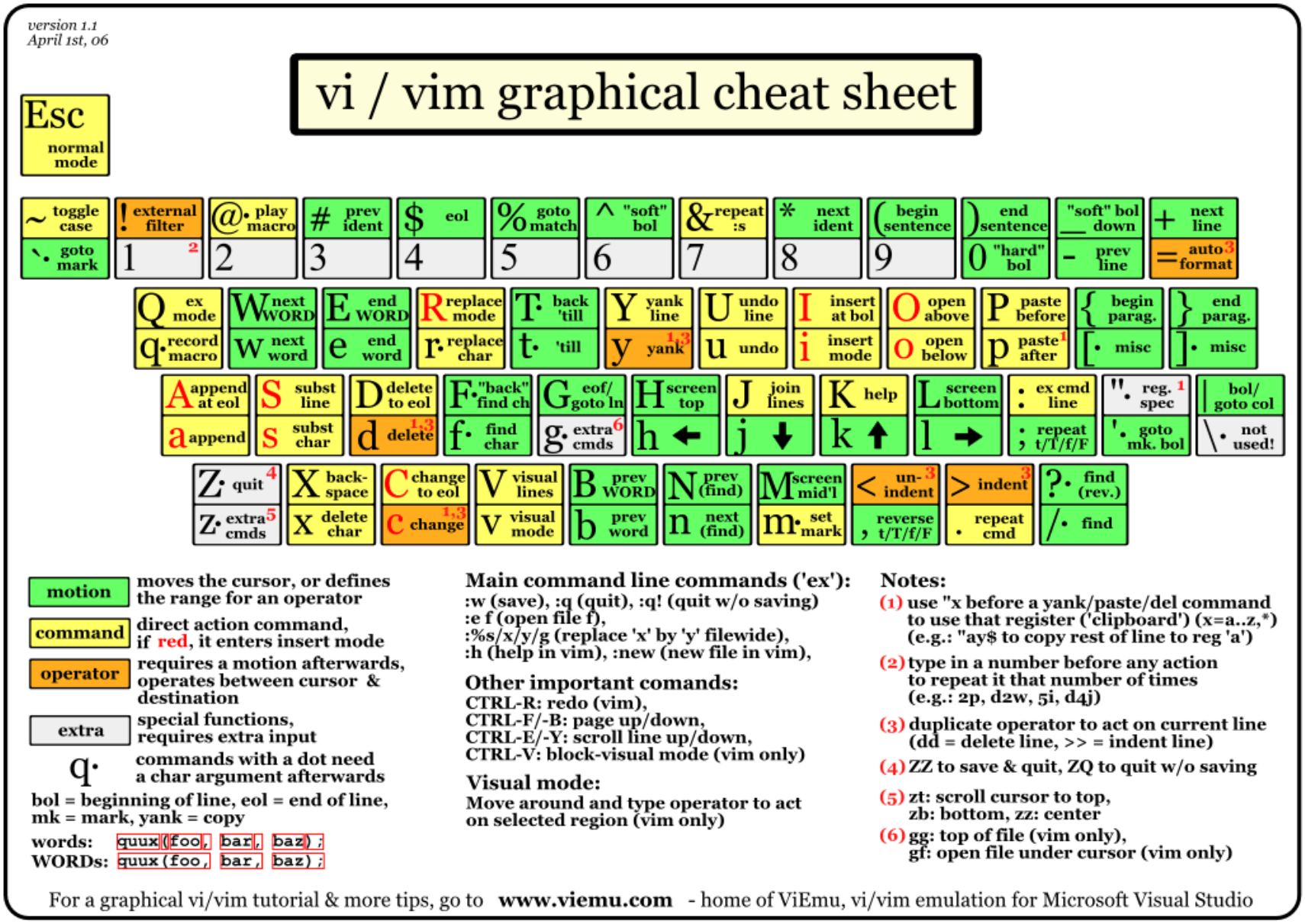
Search / Replace

/asdf - search forward for 'asdf'

- To move forward (down), type n
- To move backward (up), type N

:%s/asdf/xywz/g - replace 'asdf' by 'xywz'

Everything in a Slide



유용한 vi environment -> .vimrc

```
au FileType * setl fo-=cro
```

- 자동 주석

```
syntax on / off

set number / nonumber

set wrap / nowrap

set hlsearch

set ignorecase

set ts=4

au BufReadPost * if line("'\"")¦execute("normal `\"")¦endif
```

- 파일 오픈시 편집하던 위치로 이동

실습

** 파일 작성시 vi를 사용

- copy/paste 기능은 사용금지
- 반드시 타이핑과 편집 기능을 이용하여 작성

1. 아래의 내용을 가지는 7years.txt 파일을 작성

```
Once, I was seven years old, my mama told me
"Go make yourself some friends, or you'll be lonely"
Once, I was seven years old
It was a big-big world, but we thought we were bigger
Pushing each other to the limits, we were learning quicker
By 11, smoking herb and drinking burning liquor
Never rich, so we were out to make that steady figure
Once, I was 11 years old, my daddy told me
"Go get yourself a wife, or you'll be lonely"
Once, I was 11 years old
I always had that dream like my daddy before me
I made the man so happy when I wrote a letter once
I hope my children come and visit, once or twice a month
Soon, I'll be 60 years old, will I think the world is cold?
Or will I have a lot of children who can warm me?
Soon, I'll be 60 years old
Soon, I'll be 60 years old, will I think the world is cold?
Or will I have a lot of children who can hold me?
Soon, I'll be 60 years old
Once, I was seven years old, my mama told me
"Go make yourself some friends, or you'll be lonely"
Once, I was seven years old
Once, I was seven years old
```

실습

2. 아래의 내용을 가지는 599.c 파일을 작성

```
#include <stdio.h>
int main()
{
        printf("5*1 = 5\n");
        printf("5*2 = 10\n");
        printf("5*3 = 15\n");
        printf("5*4 = 20\n");
        printf("5*5 = 25\n");
        printf("5*6 = 30\n");
        printf("5*7 = 35\n");
        printf("5*8 = 40\n");
        printf("5*9 = 45\n");
}
```

GCC 기본 사용법
프로그램 실행법

GCC Option(1)

-o

- 출력(output) 화일명을 정하는 옵션
- gcc -o hello hello.c
  hello.c라는 파일을 hello라는 실행파일로 만듦

-l

- 필요 라이브러리를 지정

-g

- 컴파일된 오브젝트파일에 디버깅 코드를 추가
- 추후 gdb를 사용할 때 꼭 필요
- gcc -g -o hello hello.c

GCC Option(2)

-O

- 코드를 최적화(optimization)
- -O2는 가장 많이 최적화
- -O0라고 하면 최적화를 하지 않음
- 기본값: -O1 (시스템 마다 상이함)

-I

- #include 문장에서 지정한 헤더 화일이 들어있는 곳을 정하는 옵션
- gcc -c source.c -Iinclude
  -I를 붙여씀

-L

- 그 라이브러리가 어느 디렉토리에 있는지 알려 줌

Ex: vi로 c code를 작성

실습목표

- vi를 이용하여, 99단을 출력하는 프로그램을 작성
- main.c 에서는 printgugu 함수를 call
- gugu.c 에서는 printgugu를 구현하여 99단을 출력
- Compile: gcc -o a.out main.c gugu.c
- ./a.out 으로 실행

main.c	gugu.c

main() {
 	
printgugu()
 	
}
	printgugu() {
	
	do... print gugu
	
	}

Ex: vi로 c code를 작성

실습목표

- vi를 이용하여, $1^2 + 2^2 + 3^2 + \ldots + n^2$ 을 출력하는 프로그램을 작성
- main.c 에서는 n을 입력받고 calculate 함수를 call
- calc.c 에서는 calculate를 구현하여 결과를 리턴
- Compile: gcc -o b.out main.c calc.c
- ./b.out 으로 실행

main.c	calc.c

main() {
 	
calculate(n)
 	
}
	calculate(n) {
	
	do... something
	
	}