

잘생긴 상훈오빠랑 함께하는 빔 테마미나 #1

lucidash - 김상훈

범?
범?

Vim!

Vim 이 뭔가요?

Vi IMproved

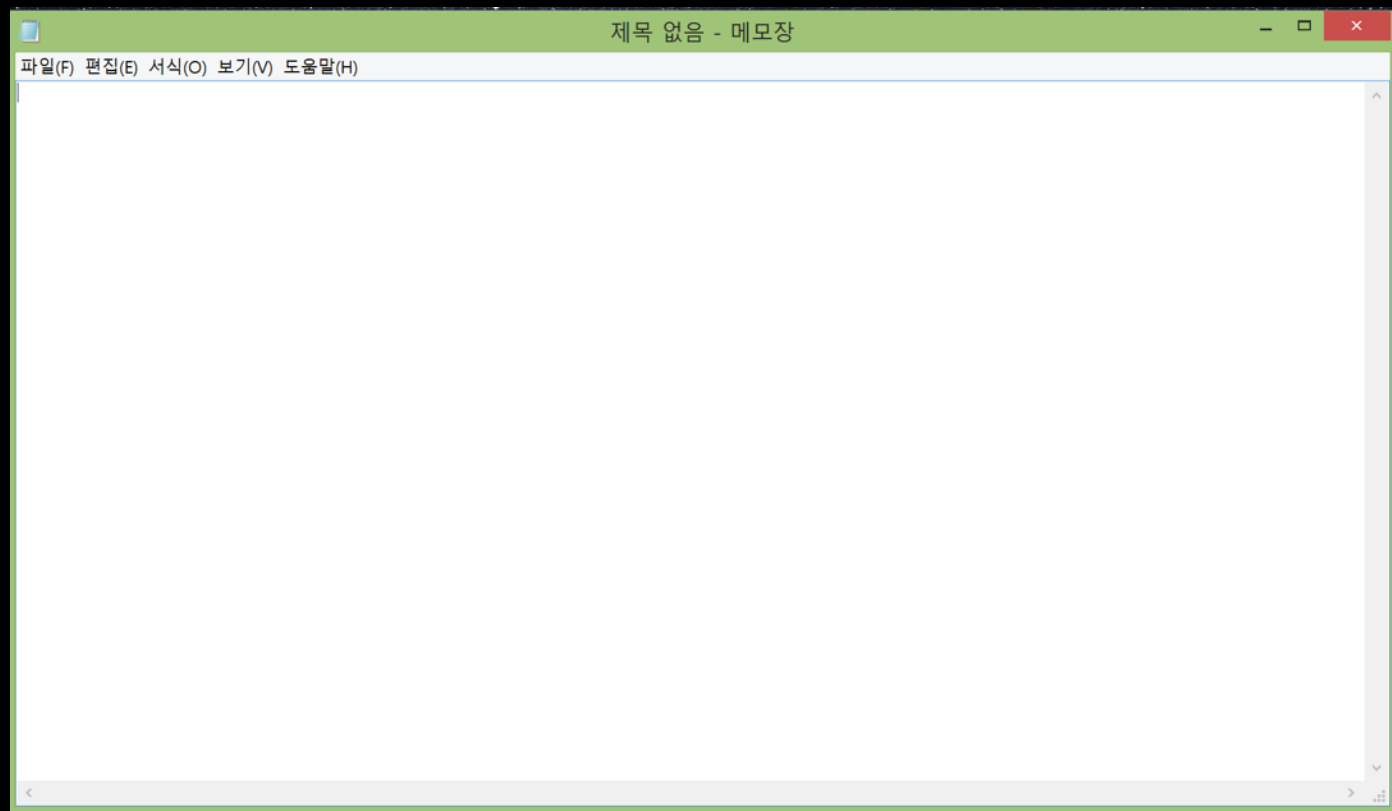
Vi 는 그림 또 뭐야?

Visual Editor

Text Editor

문서편집기

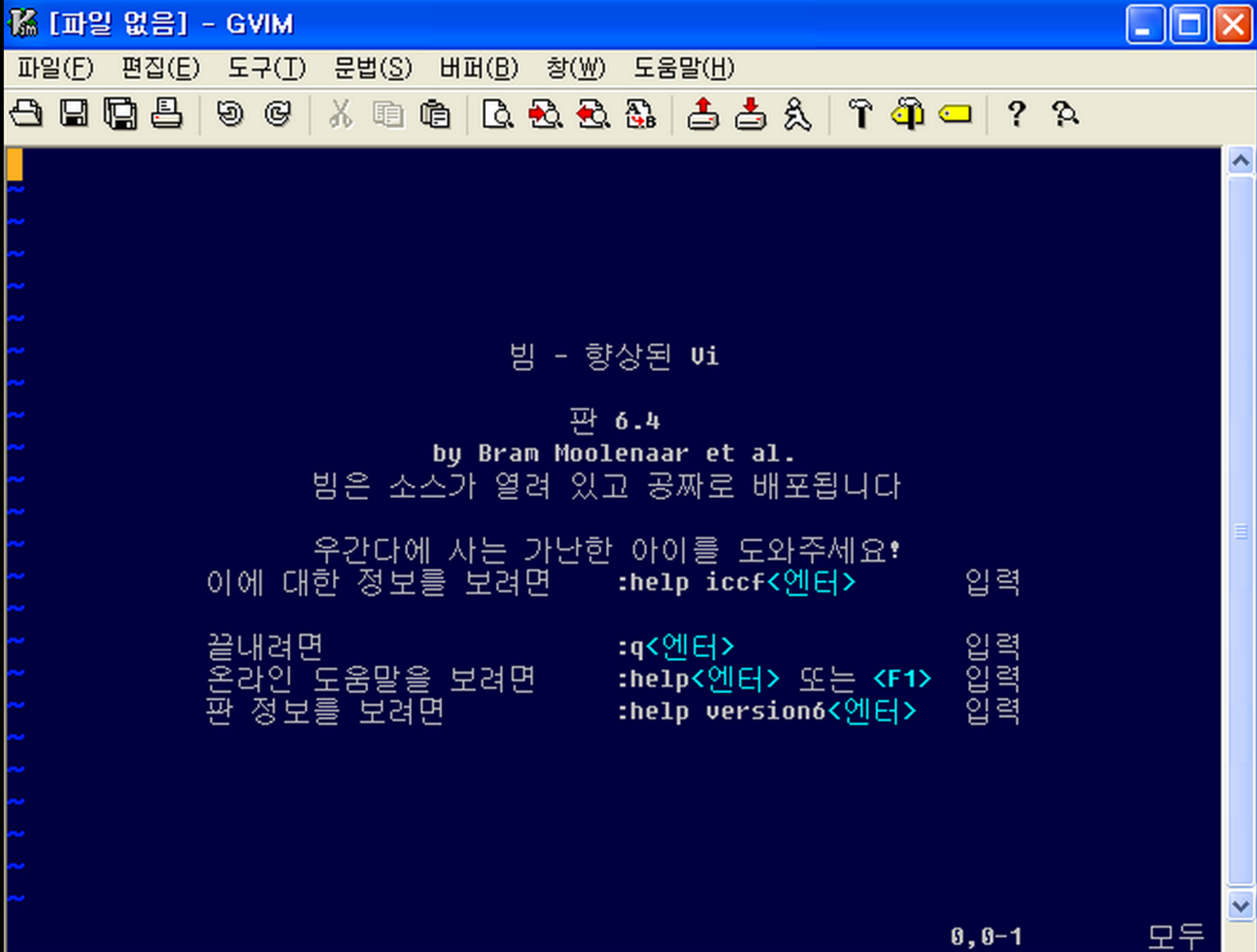
Do you know Notepad
(A.K.A 메모장) ?



간단히, Unix 기반 OS의
'메모장' 같은 녀석

그렇긴한데

gVim , MacVim...
Vim on GUI OS!!!



Awesomeful 하다

왜?

써보면 안다!!!

안 쓸 수가 없다

Vim 이란 녀석...
알아야 하는 .. 이유는 ??

Terminal 환경에서

각종 config files를 편집할 일이 생길 수 밖에 없는데
이 때 가장 간편하고 쉽게 사용할 수 있는 Text Editor
가 Vim 이라고 볼 수 있다!!!! 이 때 까지 한 페이지에
한 줄 밖에 없었으니까 읽기 어려운 페이지도 하나 째

실행!

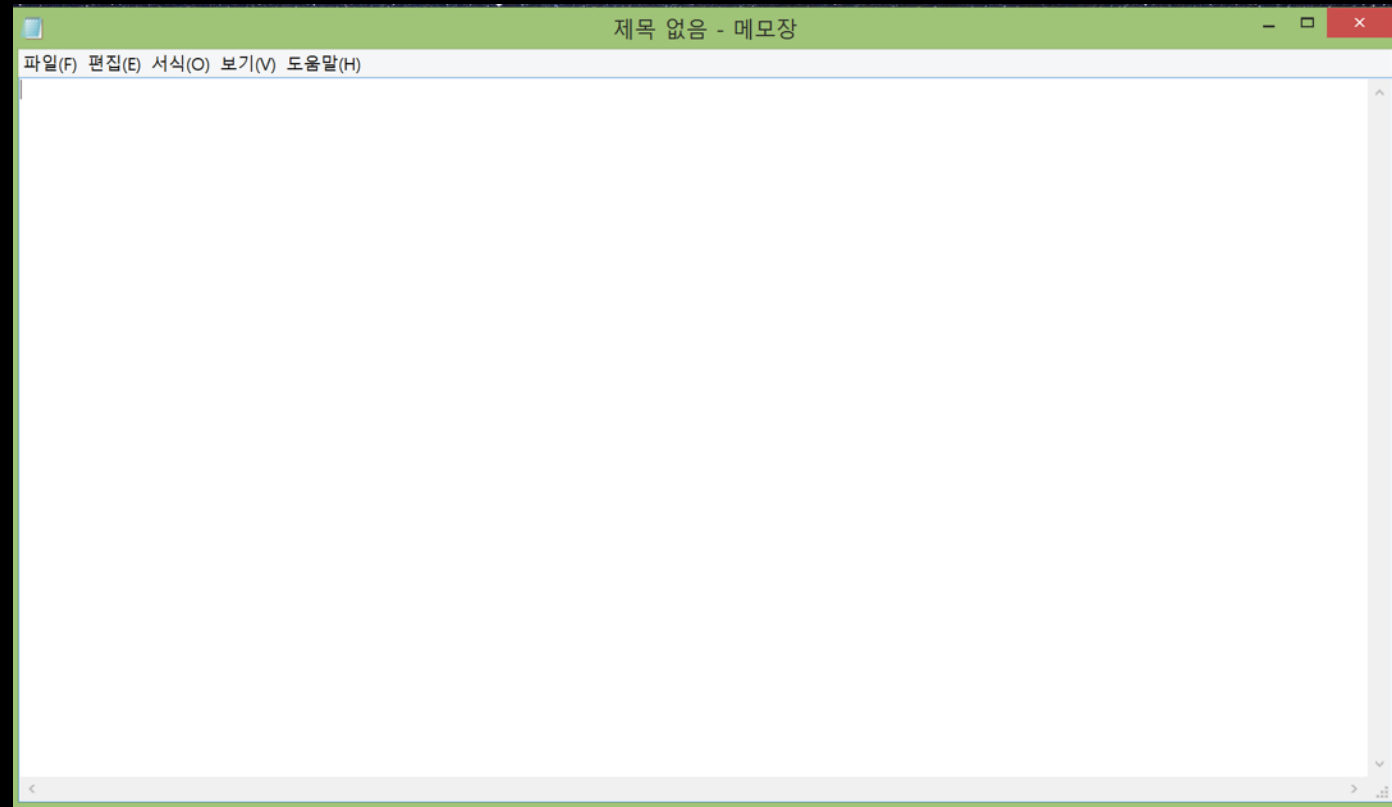
\$vim [filename]



참 쉽죠?

텍스트 입력!

Notepad ≡ ...



저기서 그냥 아무거나 막 두드리면 .. ?



Vim 은 ... ?!

Mode에 따른 Actoin!

Insert Mode

키보드 타이핑이 가능!

Nommal Mode

각종 편집질(!?) 가능

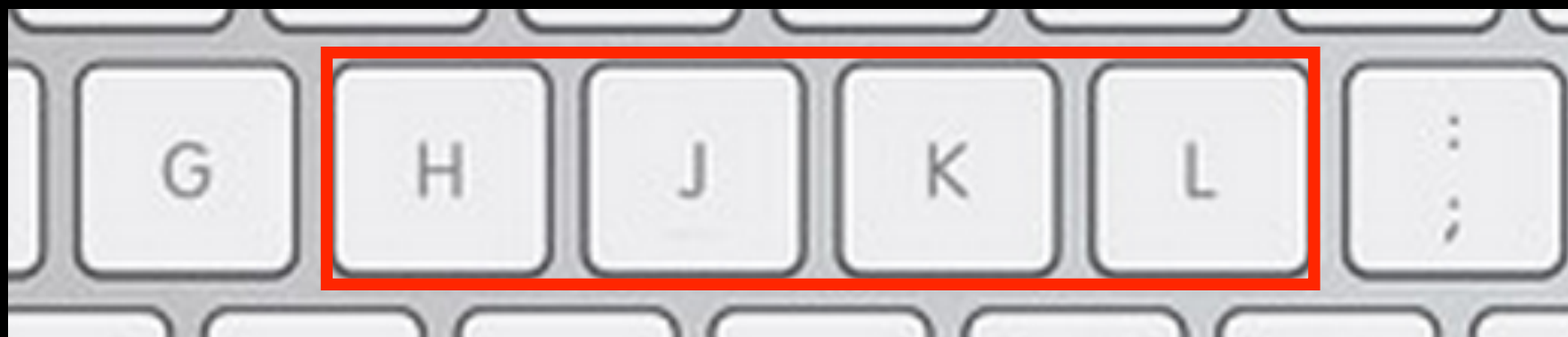
Vim 을 처음 실행하면
Normal Mode

Insert Mode 에서
[ESC] or [Ctrl - c]

ex) 커서이동 without 방향키

How ?

hijkl



h : 커서를 왼쪽으로 한 칸 이동

| : 커서를 오른쪽으로 한 칸 이동

j : 커서를 한 줄 아래로 이동

k : 커서를 한 줄 위로 이동



근데 ... 커서를 한 칸 씩만
이동 할 수 있는거예요?

그럴리가 ...

- w : 커서를 다음 단어의 첫 글자로 이동
- b : 커서를 현재 단어의 첫 글자 / 이전 단어의 첫 글자로 이동
- Ctrl - d : 1/2 page down
- Ctrl - u : 1/2 page up
- Shift - h (H) : 커서를 현재 보이는 화면의 가장 첫 줄로 이동
- Shift - l (L) : 커서를 현재 보이는 화면의 가장 아랫 줄로 이동
- Shift - 6 (^) or O(숫자영)
: 커서를 현재 라인의 가장 처음으로 이동
- 두 명령이 뭘 차이가 있을까?
- Shift - 4 (\$) : 커서를 현재 라인의 가장 끝으로 이동

- gg : 커서를 현재 파일의 가장 첫 라인으로 이동
- Shift - g (G) : 커서를 현재 파일의 가장 끝 라
인으로 이동
- # : 커서가 위치한 단어를 커서 이전 방향으로 검
색 후 검색 한 단어의 맨 앞으로 커서 이동
- * : 커서가 위치한 단어를 커서 이후 방향으로 검
색 후 검색 한 단어의 맨 앞 으로 커서 이동

그럼 ... 단어 찾으려면
단어를 직접 찾아서
#, * 을 이용해야하나요?

그럴리가 ...

/pattern

현재 문서에서 pattern을 찾음
ㅋ

찾고나서
 n, N 을 돌려보면?

n : 다음찾기
N : 이전찾기

으 ... 노답 ——

커서 움직이는거 밖에 음슴?

그럴리가 ...

- dd : 현재 라인을 삭제하고 클립보드(버퍼)에 복사
= 잘라내기 = (notepad의) ctrl - x
- yy : 현재 라인을 클립보드(버퍼)에 복사
= (컨트롤씨하는 그거)
- p : 클립보드(버퍼)에 복사 된 내용을 붙여넣기
= (컨트롤 브이브이브이 하는 그거)
- u : undo(실행 취소)
- Ctrl - r : redo(실행 취소 취소)
- Shift - j (J) : 현재 라인의 개행문자(\n, newline) 삭제

%d + 명령어!

2dd : ???

4dd : ???

9yy : ???

15j : ???

- %d [h|j|k|l] : %d 칸 만큼 h, j, k, l 를 실행!
- %d [w|b] : %d [다음|전] 단어의 첫 글자로 이동
- %d yy : %d 라인 복사
- %d dd : %d 라인 삭제
-

How to enter the
Insert Mode ?

- i : 현재 커서 위치 앞에서 insert mode
- a : 현재 커서 위치 뒤에서 insert mode
- o : 현재 라인 다음 라인에서 insert mode

- I : ? ? ? ?

- A : ? ? ? ?

- O : ? ? ? ?

- 한 번씩 해봅시다

Visual Mode

간단하게 설명하자면...

블록 지정!!!

Normal Mode에서
v(or V)를 누르면!?

v: 현재 커서 부터 블록 지정!
V: 현재 라인 블록 지정!

Visual Mode
커서이동은?

당연히 h j k l

or

w, b,

Visual Mode에서
블럭을 잡고서 복사,
삭제하려면?

Quiz .

Normal Mode 상태에서
ggVGd를 입력하면 어떻게 될까요?

Command Line

Normal Mode 상태에서
colon = ':'

각종 명령어를 실행!

실행 후 에는 Normal Mode로

- :w
- 저장한다 (ctrl + s 가 아님)
- :w!
- 강제로 저장한다
- :w filename
- filename 으로 저장한다

- :q
- 끈다(quit)
- :wq
- 저장하고 끈다
- :q!
- 강제로 끈다 (열린 파일의 변경사항을 저장하지 않음)

- :e(open) filename
 - filename 파일을 연다
- :e!(open!) filename
 - filename 을 강제로 연다

- `:help`
 - 모르는거 있을 때 쳐보자
- `:set`
 - 다양한 옵션들을 설정한다
- `:%d`
 - %d 라인으로 커서를 이동

:set 으로는 어떤 옵션들을
설정할 수 있는가? ? ?

- `:set nu`
 - number of line. 줄 번호 표시
- `:set [nonu|nu!]`
 - no number of line. 줄 번호 삭제
- `:set ai`
 - auto indent. 자동 들여쓰기 사용
- `:set [noai|ai!]`
 - 하나를 가르쳐주면 열을 알자
- `:set ft= filetype([cpp|python|java···])`
 - 현재 문서의 file type을 변경

- 이거 외에도 사실 엄청 많은데
- 하나씩 필요한 옵션이 있을 때 찾아쓰는게 이득!

음... find & replace 는 ?

: [range]s/old_pattern/
new_pattern/option

: [range]s/old_pattern/
new_pattern/option

[Range] = 범위
s = substitute

- % : 문서 전체
- 10,20 : 10번째 줄 ~ 20번째 줄 까지
- -3,+3 : 커서 기준으로 3줄 이전 부터 3줄 이후 까지
- '<,>' : 블록 설정한 내부
 - 이걸 visual mode 에서 블록잡고서 :를 입력하면 자동입력 됨

: [range]s / old_pattern /
new_pattern / option

: [range]s/old_pattern/
new_pattern/option

다른 편집기가 그렇듯,
Regular Expression 가능

: [range]s/old_pattern/
new_pattern/option

옵션도 다양하다

- [cgil]
- c : (confirm) - 각 substitution 대해 y/n?
- g : 각 줄의 모든 occurrence 를 치환
- g옵션이 없으면 각 줄의 첫 occurrence 만 치환
- i / I : case insensitive / sensitive

‘/’는 여덟해 어떻게 입력하나요?

‘V’

숙제

'hl' 100번하기
'jk' 100번하기

쓰다가 잘 모르겠으면...

[http://vimdoc.sourceforge.net/
html/doc/](http://vimdoc.sourceforge.net/html/doc/)

or

:help

Vim Seminar#2

할거임!