



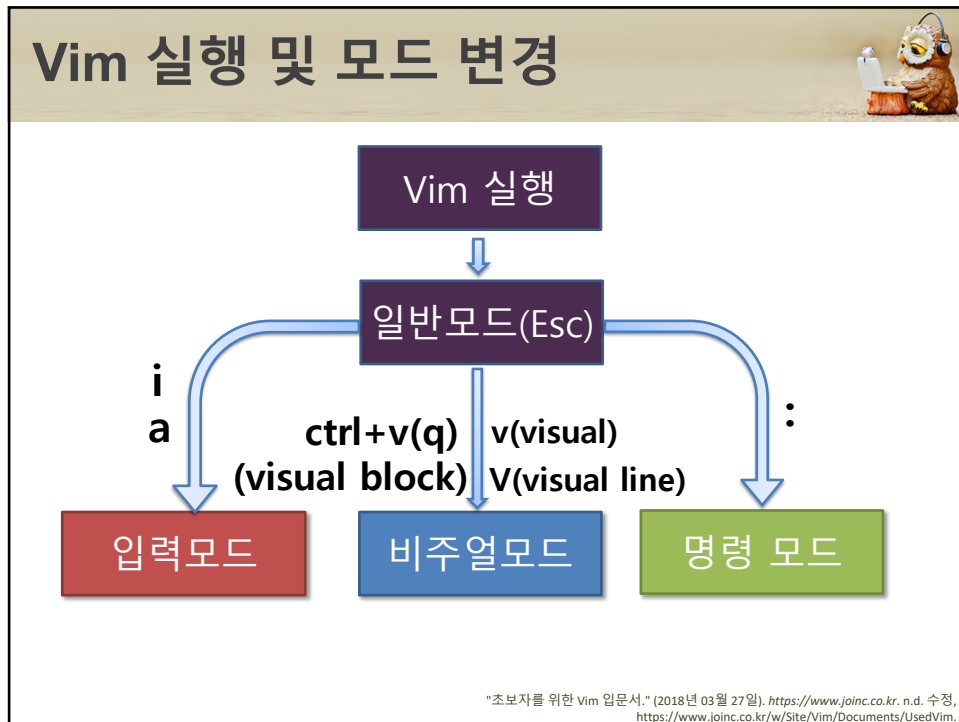
## Vim 이란?

**VIM(빔, Vi IMproved) 은?**

Bram Moolenaar가 만든 vi 호환되는 설계가 잘 된 **텍스트 편집기**이다.

**Vim 의 모드**

- 일반 모드(Normal mode)
- 명령 모드(Command mode)
- 편집/입력 모드(Insert Mode)
- 비주얼/선택 모드(Visual Mode)



## Vim 파일 생성 입력/저장

```

$ vim example.dat
i 입력 -- INSERT --
peace.handong.edu - PuTTY
Hello
[Esc] 클릭하여 일반 모드로 전환
ZZ
:wq
:w newfile
:q! :quit

```

example 파일 생성

일반모드 -> 입력모드로 변경

파일 저장 후 종료

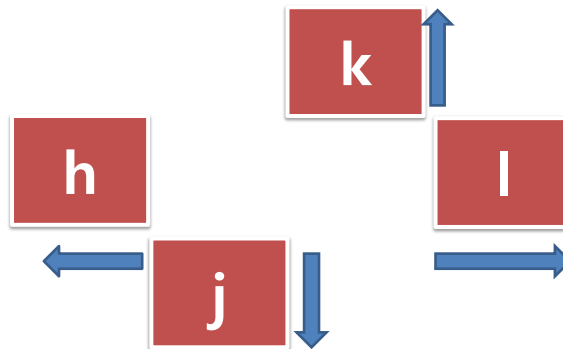
새 파일(newfile) 이름으로 저장

저장 없이 종료

# VIM – 커서 이동키(일반모드)



[Esc] 클릭하여 일반 모드로 전환 후!



# VIM – 단축키 모음



version 1.2  
April 1st, 06

## vi / vim 단축키 모음

**Esc**  
명령 모드

← 대소문자 전환	! 외부 명령	@ 매크로 실행	# 이전 검색	\$ 줄종료로 이동	% 줄종료로 %로 이동	^ 줄의 첫 글자	& 줄의 &로 이동	* 다음 검색	( 문장 시작	) 문장 끝	아래줄로 이동	+ 다음 줄
1	2	3	4	5	6	7	8	9	0	.	-	=
Q 실행 모드	W word 단어로	E word 끝으로	R 수정 모드	T word 끝에	Y 줄 전체를 복사	U 줄 전체를 취소	I 줄 시작에서 삽입	O 행 아래에 삽입	P 행 위에 삽입	{ 문장 시작	}	}
q 대소문자 전환	w 단어로	e word 끝으로	r 한 문자 교체	t 한 문자 복사	y 줄 전체 복사	u 줄 전체 취소	i 줄 시작에서 삽입	o 행 아래에 삽입	p 행 위에 삽입	[ 문장 시작	]	}
A 줄 끝에	S 줄 전체를 변경	D 삭제	f, F, t, T 한 문자로	g, G 한 줄로	H, h 줄 시작	J, j 줄 끝	K, k 줄 이동	L, l 줄 이동	.	~ 대소문자 전환	"	영역 이동
a 문장 끝	s 줄 전체를 변경	d 삭제	f, F, t, T 한 문자로	g, G 한 줄로	H, h 줄 시작	J, j 줄 끝	K, k 줄 이동	L, l 줄 이동	.	~ 대소문자 전환	"	영역 이동
Z 종료	X 줄 전체 삭제	C 줄 전체를 복사	V 줄 전체를 복사	B word 단어로	N, n 다음으로	M, m 줄 이동	< 이전으로	> 다음으로	?	찾기	/	찾기
z 종료	x 줄 전체 삭제	c 줄 전체를 복사	v 줄 전체를 복사	b word 단어로	n, N 다음으로	m, M 줄 이동	< 이전으로	> 다음으로	?	찾기	/	찾기

**동작** 커서를 이동하거나, 텍스트를 조작할 방법을 지정합니다.

**명령** 바로 동작하는 명령.

**연산자** 명령이 한 번만 실행되거나, 커서 이동과 함께 사용되어야 하며, 커서의 위치부터 목적지까지 안전합니다.

**확장** 기본에 더 추가하는 것입니다.

**q** 입력 후 (소자를 제외한) 으로 끝낼 수 있는 글자를 입력하여 합니다.

**words:** 구분자로 글자, 특수기는 모두 사용

**WORDS:** quux(foo, bar, baz) z

**words:** quux(foo, bar, baz) z

**WORDS:** quux(foo, bar, baz) z

vi/vim 에 대한 더 많은 강좌나 팁을 얻으려면 [www.viemu.com](http://www.viemu.com) (ViEmu, MS 비주얼 스튜디오를 위한 vi/vim 에뮬레이션)을 방문하십시오.

**주요 명령행 명령 ('ex'):**

- sw (지정), se (종료), sh (지정하지 않고 종료)
- se (마지막 열 끝), sh (마지막 열 끝)
- sw/A/y/ (대문자에서 's'를 'y'로 교체), sh (vim 도움말), new (새 파일)

**그 외 중요한 명령들:**

- CTRL-R: 재실행 (vim)
- CTRL-S: 새 파일로 저장 (vim)
- CTRL-E: 줄 스크롤 위로/아래로
- CTRL-V: 블록이수입 모드 (vim 전용)

**비주얼 모드:**

- 커서를 움직여 지정된 범위에 연산자를 적용합니다. (vim 전용)

**참고:**

- (1) 복사/붙여넣기/자르기 명령어를 사용하기 전에 "q"를 입력하여 레지스터(클립보드)를 지정하세요. (h)는 현재 커서는 "q"를 사용할 수 있지만 (h)는 "q"를 사용하지 않습니다. (h)는 현재 커서에서 라인 끝까지의 내용을 레지스터 'a'에 저장합니다.
- (2) 어떤 명령을 입력하기 전에 횡수(행 번호)를 지정하면, 횡수만큼 번역하게 됩니다. (예: 2p, 42w, 5l, 6d)
- (3) vi/vim으로 입력하는 명령은 현재 커서의 위치에 반영됩니다. (예: 6d(현재 라인 지우기), >>(들여쓰기))
- (4) ZZ는 저장후 종료, ZQ는 저장하지 않고 종료.
- (5) zt: 커서가 위치한 곳을 제일위로 올리거나, zb: 바닥으로, zz: 가운데로
- (6) gx: 화면의 처음으로(Vim 전용), gz: 커서가 위치한 곳의 제일 밑으로(Vim 전용)

\*"초보자를 위한 Vim 입문서." (2018년 03월 27일). <https://www.joinc.co.kr>. n.d. 수정, <https://www.joinc.co.kr/w/Site/Vim/Documents/UsedVim>

## Vim – 입력모드(Insert Mode) 진입



[Esc] 클릭하여 명령 모드에서 입력 모드로!

- **i** 현재 커서의 앞에서 편집 시작
- **a** 현재 커서의 다음부터 편집 시작
- **A** 현재 커서 줄의 맨 끝으로 이동하고 편집
- **o** 현재 커서 아래에 새로운 줄 추가하여 편집
- **O** 현재 커서 위에 새로운 줄을 추가하고, 편집

## Vim – 삭제 명령



[Esc] 클릭하여 명령 모드로 전환

- **x** 한 글자 지우기
- **dw** 현재 커서부터 단어의 끝까지 지우기
- **dd (or 3dd)** 해당 줄 전체를 지우기 위해 사용(횟수 지정시 횟수만큼)

## Vim – 취소(UNDO/REDO) 명령



[Esc] 클릭하여 명령 모드로 전환

- **u**            마지막 명령 취소
- **U**            해당 줄 전체의 수정사항 취소
- **Ctrl+r**      redo 기능

## Vim – Put, Replace 명령



[Esc] 클릭하여 명령 모드로 전환

- **p**            마지막 지운 정보를 현재 커서 다음 위치에 붙여넣기
- **r**            글자 하나를 변경할 때 사용
- **cw**          단어의 일부나 전체를 변경할 때 사용(해당 단어 삭제 후 입력)
- **c\$**          해당 줄 전체를 변경할 때 사용(전체 줄 삭제 입력)

## Vim – 파일 내 위치 이동 명령



[Esc] 클릭하여 명령 모드로 전환

- **Ctrl+g** 현재 커서의 위치와 파일 상태를 볼 수 있음
- **Shift+G** 파일에서 마지막 라인 이동
- **줄번호 + SHIFT+G** 해당 줄번호로 이동

## Vim – 찾기 명령



[Esc] 클릭하여 명령 모드로 전환

- **/찾는문자** 찾는 문자를 파일내에서 찾을
- **?찾는문자** 찾는 문자를 파일내 역방향으로 찾을
- **n** 같은 다음 단어를 찾을 수 있음
- **SHIFT+N** 같은 단어를 역방향으로 찾을

## Vim – 열기/저장



[Esc] 클릭하여 명령 모드로 전환

- **ZZ** 현재 파일을 저장하고 VIM 종료

여러 파일 열어서 작업하기

- **:vs <filename>** 수직으로 창을 나눈 후, 해당 파일을 읽어온다.
- **:split <filename>** 수평으로 창을 나눈 후, 해당 파일을 읽어온다.
- **ctrl+ww** 다음 파일의 에디터 창으로 이동

외부 명령어 실행하기

- **:! 명령어 [enter]** 셸 명령어를 실행하여 결과를 확인할 수 있고, 엔터를 치면 다시 VIM으로 복귀

## Vim – Search and replace



[Esc] 클릭하여 명령 모드로 전환

- **:s/찾을 단어/바꿀 단어/g**  
찾는 단어를 새 단어로 한 번만 변경(g -> globally)
- **:%s/ 찾을 단어/바꿀 단어 /g**  
찾는 단어를 새 단어로 전체 파일에서 모두 변경
- **:%s/ 찾을 단어/바꿀 단어 /gc** : 사용자에게 물으며 변경
- **:%s/ 찾을 단어/바꿀 단어 /i** : 대소문자 구분 없이 검색
- **:#,#s/ 찾을 단어/바꿀 단어 /g**  
두 줄 번호(#) 사이의 찾는 단어를 새 단어로 모두 변경

## Vim – Search and replace



[Esc] 클릭하여 명령 모드로 전환

- **:s/찾을 단어/바꿀 단어/g**  
찾는 단어를 새 단어로 한 번만 변경(g -> globally)
- **:%s/ 찾을 단어/바꿀 단어 /g**  
찾는 단어를 새 단어로 전체 파일에서 모두 변경
- **:%s/ 찾을 단어/바꿀 단어 /gc** : 사용자에게 물으며 변경
- **:%s/ 찾을 단어/바꿀 단어 /i** : 대소문자 구분 없이 검색
- **:#,#s/ 찾을 단어/바꿀 단어 /g**  
두 줄 번호(#) 사이의 찾는 단어를 새 단어로 모두 변경

## Lab – vi(m) 코딩 & 컴파일



```
문자열 몇개? 5
문자열입력: apple
문자열입력: banana
문자열입력: watermelon
문자열입력: grape
문자열입력: pineapple
문장 apple
문장 banana
문장 watermelon
문장 grape
문장 pineapple
```

1. char \*\*str;
2. 입력 함수 생성
3. 목록보기 함수 생성
4. 파일저장 함수 생성
5. 파일 읽기 함수 생성