

Vim Seminar

멋쟁이사자처럼 첫 수업
2013. 04. 03 (최용철)

Vim을 왜쓰는가?

뭐든지 시작은 간지로..

자동차 운전

수영 강습

코딩은 빔

제대로 못 쓰면 최악의 에디터...
제대로 쓰면 최고의 에디터...

특징

1. 대부분의 운영체제를 지원
2. 매우 작고 빠르다

<이외의 특징들..(궁금하면 찾아보세요. 궁금해야 합니다.)>

1. 텍스트 환경에서나 콘솔환경에서도 작동한다.
2. GUI 환경도 지원한다
3. 프로그래밍 언어별특수기능을 지원한다.
4. 플러그인에 따라 다양한 확장기능을 지원한다.
5. 내장 스크립트로 원하는 기능을 만들 수 있다.(커스터마이징 등)
외부명령어와 연동할 수 있다.
6. 작업 녹화나 자동화, 배치(batch) 기능을 제공한다.

설치하기 (Windows)

설치하기 (Mac)

<실행하기>

(윈도우)

파일에 마우스 오른쪽 버튼 클릭 -> vim으로 편집

(맥)

터미널에서 'vi' 또는 'vi [파일명]'

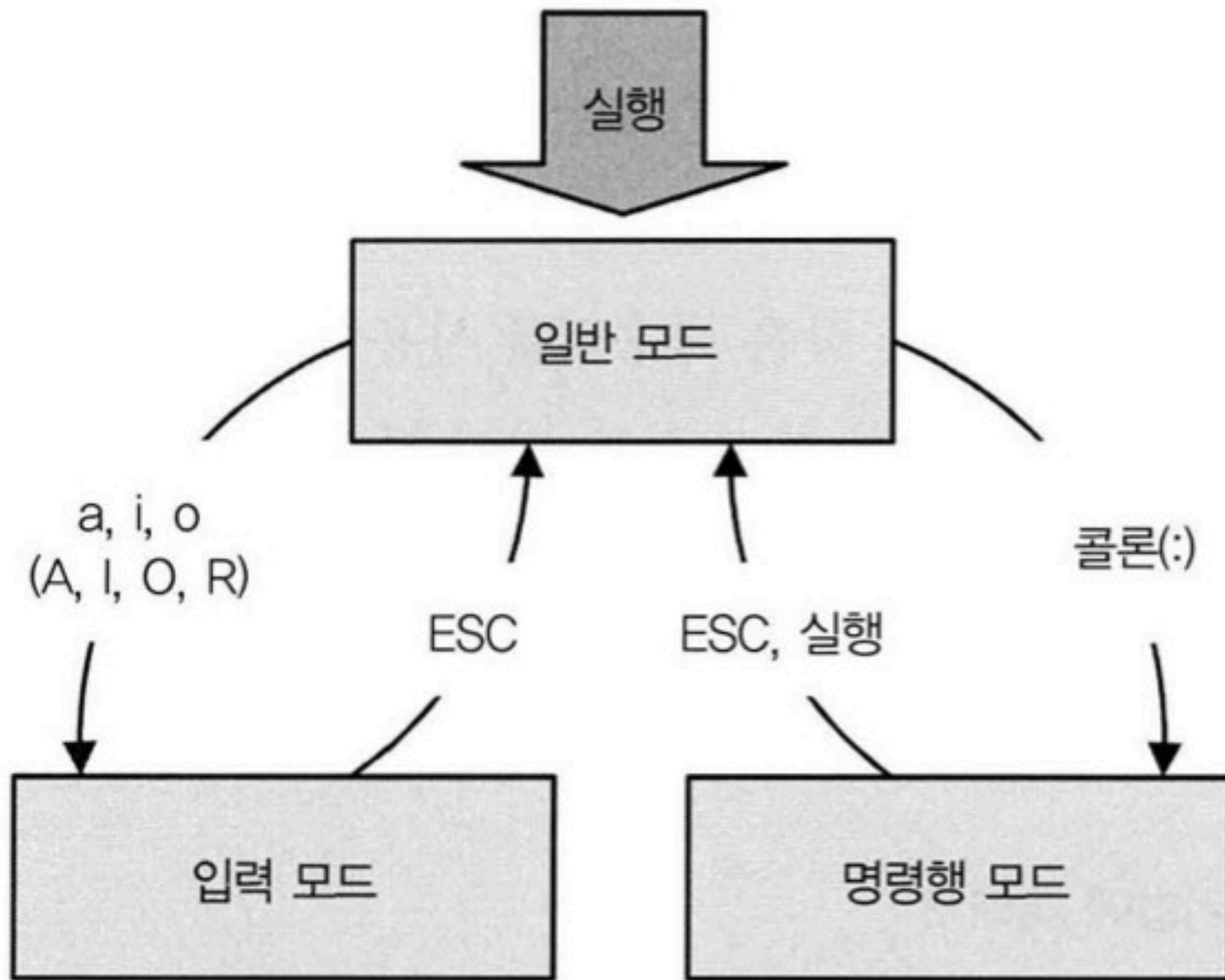


그림 2.5 vi의 세 가지 모드

Vim의 많은 단축키들.. (전부 다 쓰진 않습니다~)



| | | | |
|----------------|--|--|--|
| 동작 | 커서를 이동하거나, 연산자가 동작할 범위를 지정합니다. | | |
| 명령 | 바로 동작하는 명령입니다. 빨간색은 입력 모드로 변경됩니다. | <ul style="list-style-type: none"> 명령행 모드의 주요 명령어 <ul style="list-style-type: none"> :w 저장 :q 종료 :q! 저장없이 종료 :ef f 파일 열기 :%s/x/y/g 파일 전체에서 'x'를 'y'로 교체 :h name name 명령에 대한 도움말 :new 새 파일 일반 모드의 주요 명령어 <ul style="list-style-type: none"> CTRL-R 재실행 CTRL-F/-B 페이지 위로/아래로 CTRL-E/-Y 스크롤 위로/아래로 CTRL-V 비주얼 모드 | <ul style="list-style-type: none"> 참고 <ul style="list-style-type: none"> ① 복사/붙여넣기/지우기 명령을 사용하기 전에 "a를 입력하여 레지스터 a를 지정할 수 있습니다. (레지스터 이름은 a부터 z까지 사용 가능) 예를 들어 "ay\$는 커서 위치부터 행 끝까지의 내용을 레지스터 a에 저장합니다. ② 명령어 입력 전 숫자를 지정하면, 해당 숫자만큼 명령어가 반복됩니다. ③ 연속으로 입력하면, 현재 행에 반영됩니다. 예를 들어 dd는 현재 행이 지워집니다. ④ ZZ는 저장 후 종료, ZQ는 저장 없이 종료입니다. ⑤ z는 커서가 위치한 부분을 화면 상단으로 스크롤합니다. zb는 바닥으로, zz는 가운데로 스크롤합니다. ⑥ gg는 커서를 파일 처음으로 이동합니다. gf는 커서 위치의 파일명을 인식하여 파일을 엽니다. |
| 오퍼레이션 펜딩 모드 | 커서 위치부터 목적지까지를 대상으로 명령을 실행합니다. | | |
| 확장 | 추가적인 키 입력이 필요합니다. | | |
| q. | 입력 후 글자를 입력해야 합니다. | | |
| 단어 | 공백 문자나 특수 문자로 구분된 단어 test(123, 456, 789); | | |
| 단어 (의미상) | 공백 문자로 구분된 단어 test(123, 456, 789); | | |

일반모드(Normal Mode)에서 입력모드(Insert Mode)가기~

Esc
명령 모드

Vim 명령어 단축키

손에 잡히는 Vim 안내

| | | | | | | | | | | | | |
|-------------------------|------------------------------|----------------------------|------------------------------|------------------------------|--------------------------|-------------------------|-------------------------------|-------------------------|--------------------------------|-------------------------|-------------------|---------------------|
| ~ 대소문자 전환 , 표식으로 이동 | ! 외부 명령 1 | @. 매크로 실행 2 | # 이전 검색 3 | \$ 행 끝으로 이동 4 | % 짝 괄호 찾기 5 | ^ 행의 첫 글자 6 | & :s 반복 7 | * 다음 검색 8 | (문장 시작 9 |) 행 앞으로 0 문장 끝 | _ 아래 행 - 이전 행 | + 다음 행 = 자동 들여쓰기 |
| Q 실행 모드 q. 매크로 기록 | W 다음 단어 (의미상) w 다음 단어 | E 단어 (의미상) 끝으로 e 단어 끝으로 | R 수정 모드 r. 한 문자 교체 | T. 행에서 후방 검색 t. 행에서 전방 검색 | Y 행 단위 복사 y 복사 | U 행 단위 실행 취소 u 실행 취소 | I 행 시작에 삽입 i 커서 앞에 삽입 | O 행 위에 삽입 o 행 아래에 삽입 | P 커서 이전에 붙여넣기 p 커서 이후에 붙여넣기 | { 문단 시작 [| } 문단 끝] | |
| A 행 끝에 삽입 a 커서 뒤에 삽입 | S 삭제 후 입력 모드 s 삭제 후 입력 모드 | D 형 삭제 d 삭제 | F. 행에서 후방 검색 f. 행에서 전방 검색 | G 파일 끝으로 이동 g. 확장 명령 | H 화면 상단 h ← | J 아래 행 합치기 j ↓ | K 도움말 k ↑ | L 화면 하단 l → | : 명령행 모드 ; I/T/I/F 반복 | '' 레지스터 지정 ' 표식으로 이동 | 열 이동 \ 사용하지 않음 | |
| Z. 종료 Z. 확장 명령 | X 백스페이스 X 글자 삭제 | C 행 끝까지 바꾸기 c 바꾸기 | V 비주얼 모드 v 비주얼 모드 | B 이전 (의미상) b 이전 단어 | N 다음 (의미상) n 다음 (의미상) | M 화면 가운데 m 표식 설정 | < 내어 쓰기 , I/T/I/F 반대 방향 검색 | > 들어 쓰기 . 명령 반복 | ? 찾기 (위쪽) / 찾기 (아래쪽) | | | |

우선은...

| | |
|-------------|--|
| 동작 | 커서를 이동하거나, 연산자가 동작할 범위를 지정합니다. |
| 명령 | 바로 동작하는 명령입니다. 빨간색 은 입력 모드로 변경됩니다. |
| 오퍼레이션 펜딩 모드 | 커서 위치부터 목적지까지를 대상으로 명령을 실행합니다. |
| 확장 | 추가적인 키 입력이 필요합니다. |
| q. | 입력 후 글자를 입력해야 합니다. |
| 단어 | 공백 문자나 특수 문자로 구분된 단어 test(123, 456, 789): |
| 단어 (의미상) | 공백 문자로 구분된 단어 test(123, 456, 789): |

• 명령행 모드의 주요 명령어

:w 저장

:q 종료

:q! 저장없이 종료

:ef f 파일 열기

:%s/x/y/g 파일 전체에서 'x'를 'y'로 교체

:h name name 명령어 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

① 복사/붙여넣기/지우기 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.

② 명령어 입력 전 숫자를 지정하면, 해당 숫자만큼 명령어가 반복됩니다.

③ 연속으로 입력하면, 현재 행에 반영됩니다.
예를 들어 dd는 현재 행이 지워집니다.

④ ZZ는 저장 후 종료, ZQ는 저장 없이 종료입니다.

⑤ zi는 커서가 위치한 부분을 화면 상단으로 스크롤합니다.
zb는 바닥으로, zz는 가운데로 스크롤합니다.

⑥ gg는 커서를 파일 처음으로 이동합니다.
gf는 커서 위치의 파일명을 인식하여 파일을 엽니다.

출처 : www.viemu.com 번역 : Mr. Dust

일반모드(Normal Mode)에서 입력모드(Insert Mode)가기~

Esc
명령 모드

Vim 명령어 단축키

손에 잡히는 Vim 안내

| | | | | | | | | | | | | |
|-------------------------|------------------------------|----------------------------|------------------------------|------------------------------|------------------------|-------------------------|-------------------------------|-------------------------|--------------------------------|----------------------|-------------------|---------------------|
| ~ 대소문자 전환 , 표식으로 이동 | ! 외부 명령 1 | @. 매크로 실행 2 | # 이전 검색 3 | \$ 행 끝으로 이동 4 | % 짝 괄호 찾기 5 | ^ 행의 첫 글자 6 | & :s 반복 7 | * 다음 검색 8 | (문장 시작 9 |) 행 앞으로 0 문장 끝 | _ 아래 행 - 이전 행 | + 다음 행 = 자동 들여쓰기 |
| Q 실행 모드 q. 매크로 기록 | W 다음 단어 (의미상) w 다음 단어 | E 단어 (의미상) 끝으로 e 단어 끝으로 | R 수정 모드 r. 한 문자 교체 | T. 행에서 후방 검색 t. 행에서 전방 검색 | Y 행 단위 복사 y 복사 | U 행 단위 실행 취소 u 실행 취소 | I 행 시작에 삽입 i 커서 앞에 삽입 | O 행 위에 삽입 o 행 아래에 삽입 | P 커서 이전에 붙여넣기 p 커서 이후에 붙여넣기 | { 문단 시작 [| } 문단 끝] | |
| A 행 끝에 삽입 a 커서 뒤에 삽입 | S 삭제 후 입력 모드 s 삭제 후 입력 모드 | D 행 삭제 d 삭제 | F. 행에서 후방 검색 f. 행에서 전방 찾기 | G 파일 끝으로 이동 g. 확장 명령 | H 화면 상단 h ← | J 아래 행 합치기 j ↓ | K 도움말 k ↑ | L 화면 하단 l → | : 명령행 모드 ; i/T/I/F 반복 | 레지스터 지정 ' 표식으로 이동 | 열 이동 \ 사용하지 않음 | |
| Z. 종료 z. 확장 명령 | X 백스페이스 x 글자 삭제 | C 행 끝까지 바꾸기 c 바꾸기 | V 비주얼 라인 모드 v 비주얼 모드 | B 이전 단어 (의미상) b 이전 단어 | N 이전 (찾기) n 다음 (찾기) | M 화면 가운데 m 표식 설정 | < 내어 쓰기 , i/T/I/F 반대 방향 검색 | > 들어 쓰기 . 명령 반복 | ? 찾기 (위로) / 찾기 (아래로) | | | |

| | |
|----------------|--|
| 동작 | 커서를 이동하거나, 연산자가 동작할 범위를 지정합니다. |
| 명령 | 바로 동작하는 명령입니다. 빨간색 은 입력 모드로 변경됩니다. |
| 오퍼레이션 펜딩 모드 | 커서 위치부터 목적지까지를 대상으로 명령을 실행합니다. |
| 확장 | 추가적인 키 입력이 필요합니다. |
| q. | 입력 후 글자를 입력해야 합니다. |
| 단어 | 공백 문자나 특수 문자로 구분된 단어 test(123, 456, 789): |
| 단어 (의미상) | 공백 문자로 구분된 단어 test(123, 456, 789): |

• 명령행 모드의 주요 명령어

:w 저장

:q 종료

:q! 저장없이 종료

:ef f 파일 열기

:%s/x/y/g 파일 전체에서 'x'를 'y'로 교체

:h name 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

① 복사/붙여넣기/지우기 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.

② 명령어 입력 전 숫자를 지정하면, 해당 숫자만큼 명령어가 반복됩니다.

③ 연속으로 입력하면, 현재 행에 반영됩니다.
예를 들어 dd는 현재 행이 지워집니다.

④ ZZ는 저장 후 종료, ZQ는 저장 없이 종료입니다.

⑤ zt는 커서가 위치한 부분을 화면 상단으로 스크롤합니다.
zb는 바닥으로, zz는 가운데로 스크롤합니다.

⑥ gg는 커서를 파일 처음으로 이동합니다.
gf는 커서 위치의 파일명을 인식하여 파일을 엽니다.

출처 : www.viemu.com 번역 : Mr. Dust

일반모드(Normal Mode)에서 입력모드(Insert Mode)가기~

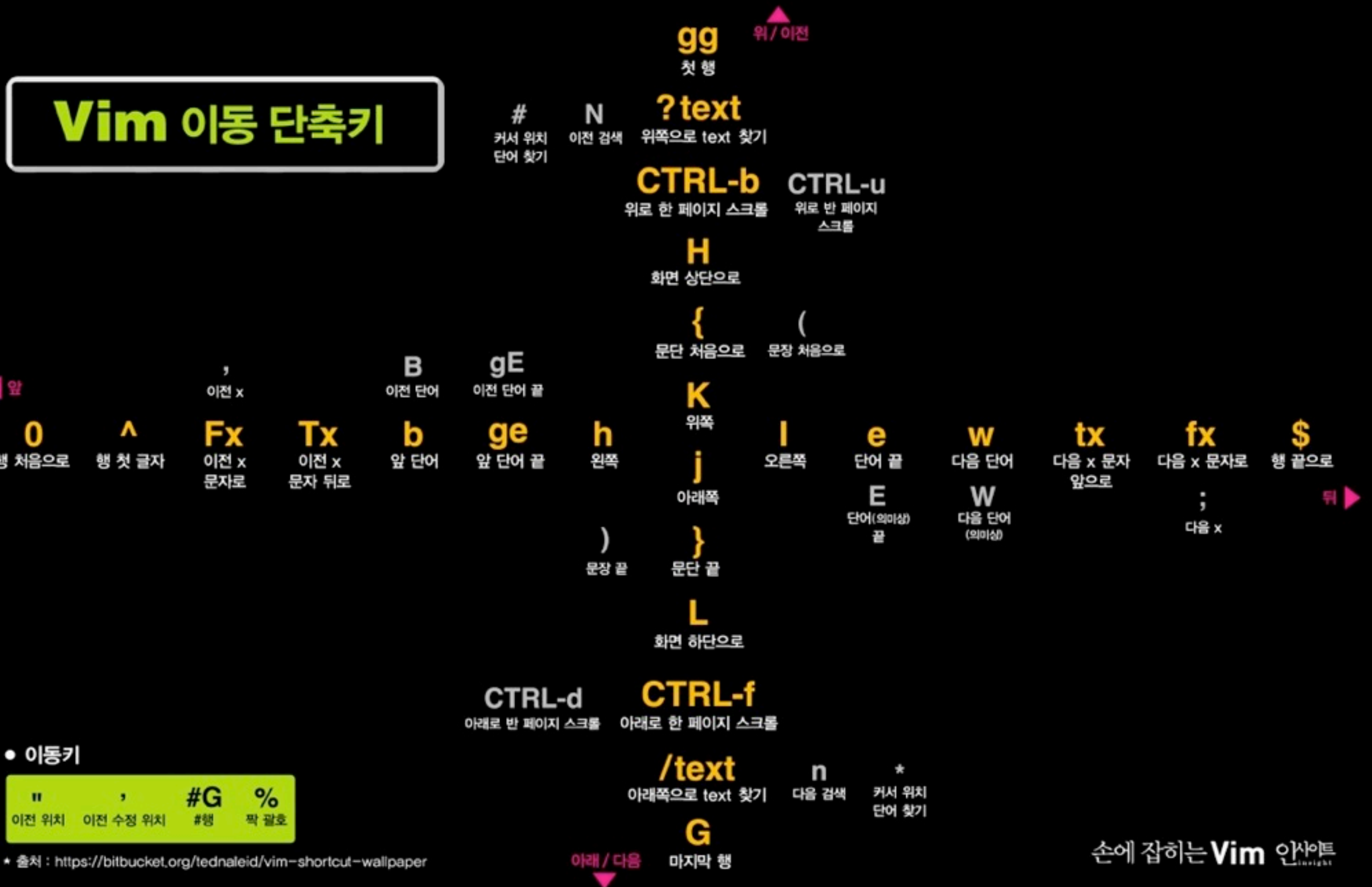


| | |
|-------------|--|
| 동작 | 커서를 움직이는 명령어입니다. 연산자가 동작할 범위를 지정합니다. |
| 명령 | 바로 동작하는 명령어입니다. 빨간색은 입력 모드로 변경됩니다. |
| 오퍼레이션 펜딩 모드 | 커서 위치부터 목적지까지를 대상으로 명령을 실행합니다. |
| 확장 | 추가적인 키 입력이 필요합니다. |
| q. | 입력 후 글자를 입력해야 합니다. |
| 단어 | 공백 문자나 특수 문자로 구분된 단어 test(123, 456, 789): |
| 단어 (의미상) | 공백 문자로 구분된 단어 test(123, 456, 789): |

- 명령행 모드의 주요 명령어
 - :w 저장
 - :q 종료
 - :q! 저장없이 종료
 - :ef f 파일 열기
 - :%s/x/y/g 파일 전체에서 'x'를 'y'로 교체
 - :h name 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
 - ① 복사/붙여넣기/지우기 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.
 - ② 명령어 입력 전 숫자를 지정하면, 해당 숫자만큼 명령어가 반복됩니다.
 - ③ 연속으로 입력하면, 현재 행에 반영됩니다. 예를 들어 dd는 현재 행이 지워집니다.
 - ④ ZZ는 저장 후 종료, ZQ는 저장 없이 종료입니다.
 - ⑤ zi는 커서가 위치한 부분을 화면 상단으로 스크롤합니다. zb는 바닥으로, zz는 가운데로 스크롤합니다.
 - ⑥ gg는 커서를 파일 처음으로 이동합니다. gf는 커서 위치의 파일명을 인식하여 파일을 엽니다.

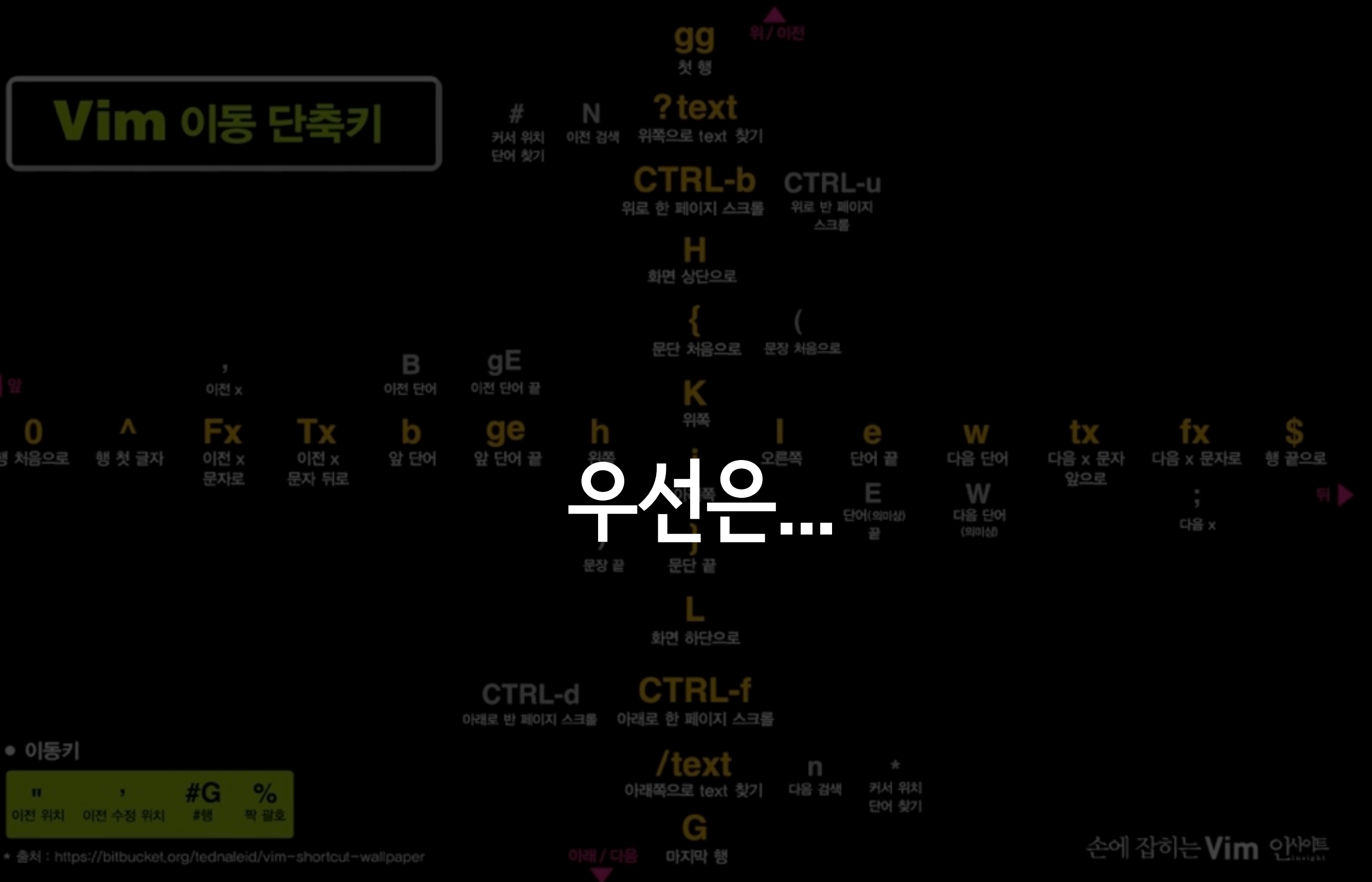
Vim 이동 단축키



손에 잡히는 Vim 인사이트

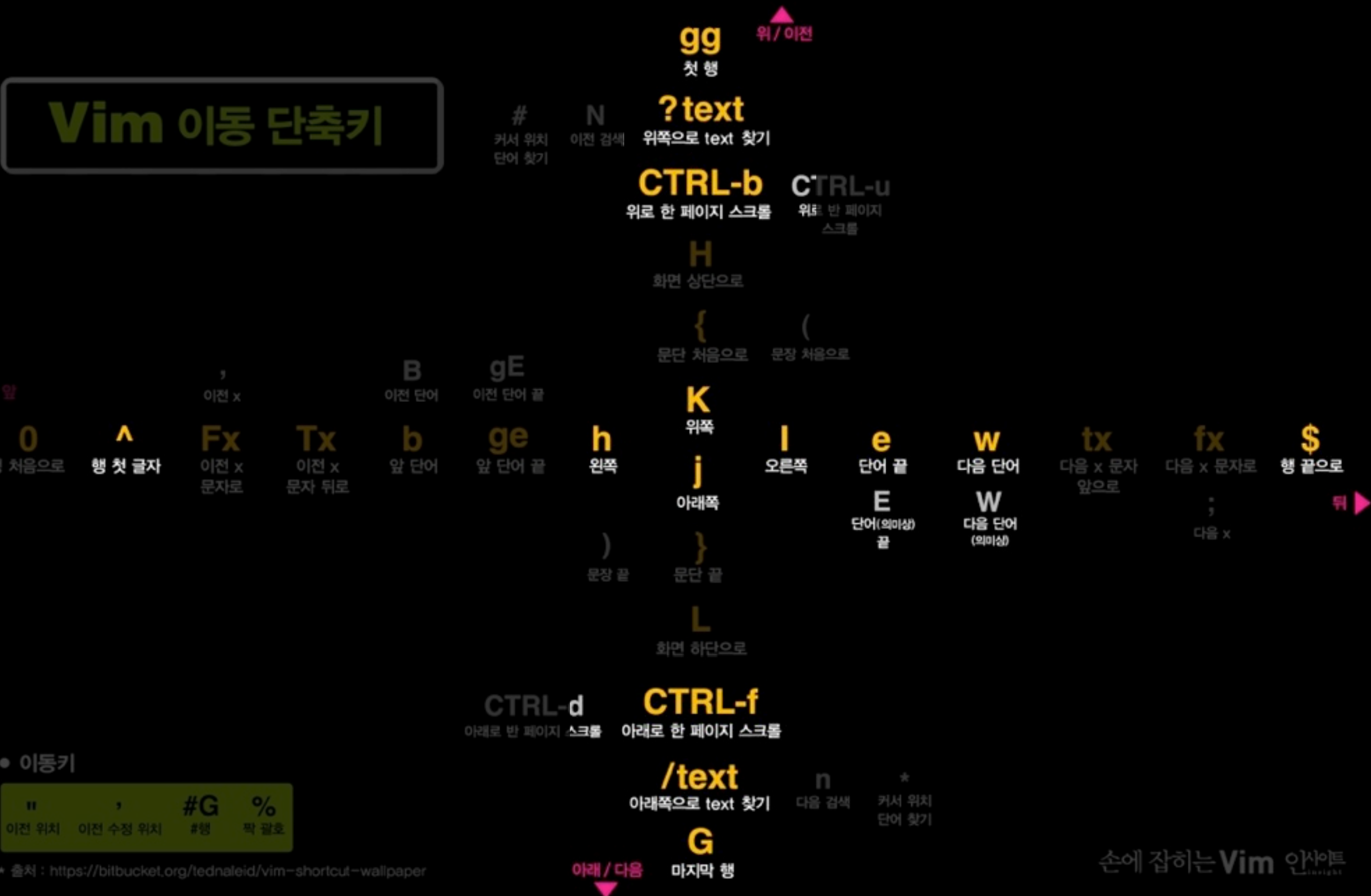
일반모드(Normal mode)에서 커서 획득 움직임

Vim 이동 단축키



일반모드(Normal mode)에서 커서 획득 움직이기

Vim 이동 단축키



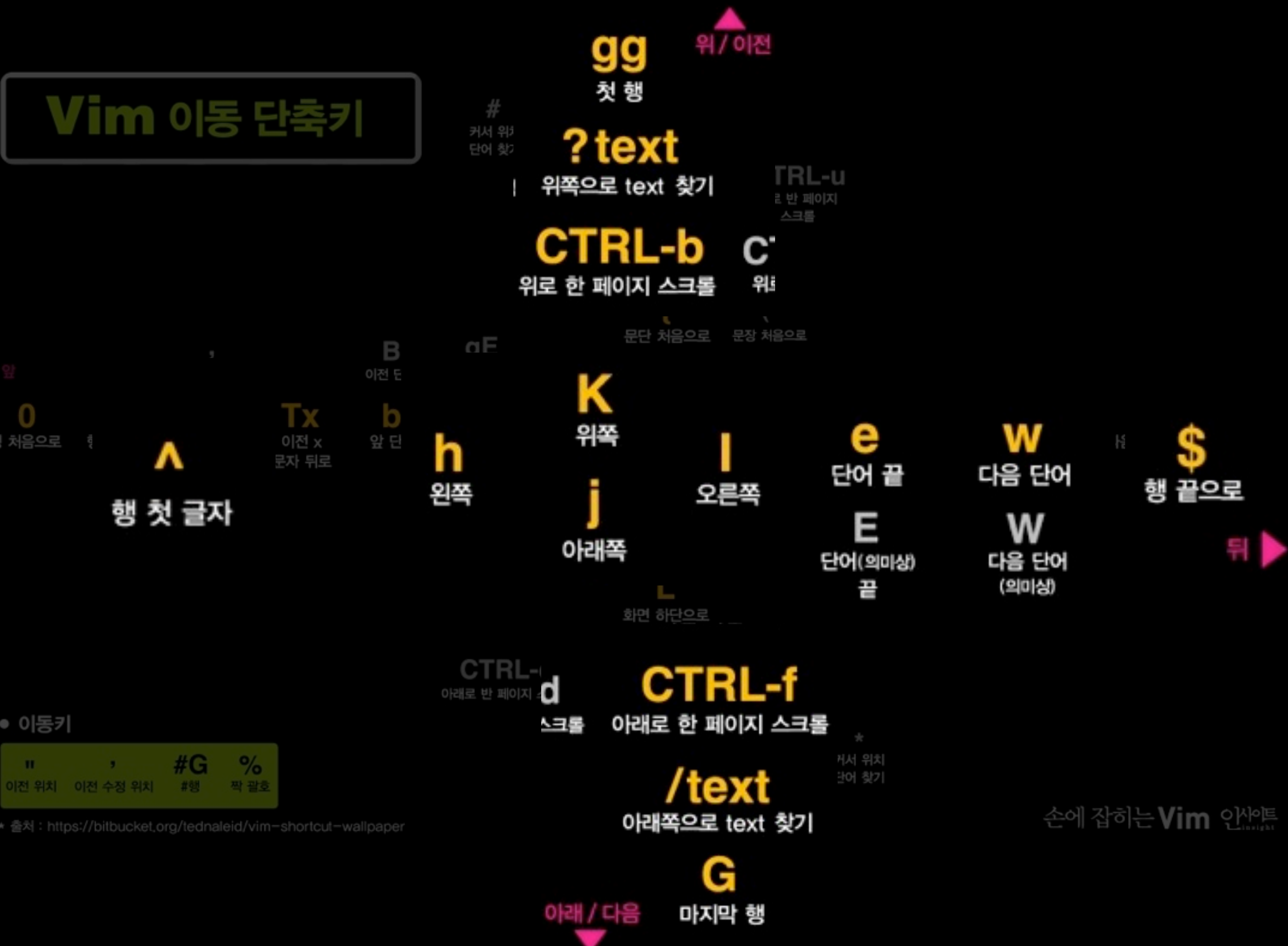
이동키

| | | | |
|-------|----------|----|------|
| " | , | #G | % |
| 이전 위치 | 이전 수정 위치 | #행 | 딱 괄호 |

* 출처 : <https://bitbucket.org/lednaleid/vim-shortcut-wallpaper>

손에 잡히는 Vim 인사이트

Vim 이동 단축키



● 이동키

| | | | |
|-------|----------|----|------|
| " | , | #G | % |
| 이전 위치 | 이전 수정 위치 | #행 | 짝 괄호 |

* 출처 : <https://bitbucket.org/tehnaleid/vim-shortcut-wallpaper>

손에 잡히는 Vim insight

vim의 dd,x(삭제) = 잘라내기

표 2.7 삭제 명령

| 명령어 | 설명 |
|-----|--------------------------------------|
| x | 커서에 위치한 문자 삭제(⟨Delete⟩ 키와 같음) |
| dd | 현재 행을 삭제 |
| :d | |
| D | 현재 컬럼 위치에서 현재 행의 끝부분까지 삭제(d\$와 동일) |
| J | 아래 행을 현재 행의 끝에 붙임(아래 행의 앞부분 공백은 제거됨) |

표 2.9 레지스터에 행을 복사하는 명령

| 명령어 | 설명 |
|---------------|--------------------------|
| yy :y Y | 현재 행을 레지스터에 복사(yank)합니다. |

표 2.8 레지스터의 내용을 문서에 붙이는 명령

| 명령어 | 설명 |
|---------------|---|
| p :pu P | 현재 행에 붙여 넣습니다(put). 개행 문자가 포함된 경우에는 현재 행의 아래에 붙여 넣습니다. 현재 행의 위쪽에 붙입니다. |

표 2.10 undo와 redo 기능

| 명령어 | 설명 |
|--------|---------------------------------------|
| u | undo 기능입니다. 바로 이전에 행한 명령 하나를 취소합니다. |
| CTRL-R | redo 기능입니다. 바로 이전에 취소했던 명령을 다시 실행합니다. |

표 2.11 명령행 모드에서 범위 지정 예

| 명령어 | 설명 |
|----------|--|
| :20d | 20번 행을 삭제합니다. |
| :10,25d | 10~25번 행을 삭제합니다. |
| :10,\$d | 10~마지막 행까지 삭제합니다. |
| :%y | 문서 전체를 복사합니다. %는 1,\$와 동일합니다. |
| :.,+20y | 현재 행부터 아래로 스무 행을 복사합니다. |
| :-10,+5d | 현재 행부터 위로 10행, 아래로 5행, 총 열여섯 행을 삭제합니다. |
| :40pu | 40번 행에 레지스터의 내용을 붙여넣습니다. |

표 3.1 3장에서 살펴볼 기능

| 살펴볼 기능 | 명령어 |
|--------|--------------------|
| 옵션 설정 | :set ... |
| 옵션 파일 | .vimrc (혹은 _vimrc) |

표 3.2 set 명령어 사용법

| 명령어 | 설명 |
|-----------------|---|
| :set | 현재 옵션 설정을 보여줍니다. |
| :set all | 모든 옵션 설정을 보여줍니다. (off 상태의 옵션까지 모두 출력) |
| :set [no]name | name에 해당하는 옵션을 켜거나 끕니다. 앞에 no를 붙이면 off 상태가 됩니다. |
| :set name! | name 옵션의 on, off 상태를 전환(toggle)합니다. |
| :set name=value | name에 해당하는 옵션에 value 값을 할당합니다. name만 지정하면 해당 옵션의 value 값을 보여줍니다. |

표 3.3 편집에 관련된 기초 옵션들

| 옵션 이름(괄호 안은 긴 이름) | 설명 |
|-----------------------|--------------------------------|
| nu (number) | 화면에 행 번호를 표시합니다. |
| ai (autoindent) | 자동 들여쓰기를 사용합니다. |
| cindent | C 언어 스타일의 들여쓰기를 사용합니다. |
| ts=value (tabstop) | 탭 크기를 value로 지정합니다. |
| sw=value (shiftwidth) | 블록 이동시 열의 너비입니다. |
| tw=value (textwidth) | 편집할 화면의 열 길이를 정합니다. (0이면 비활성화) |


```
1 call pathogen#infect()
2 syntax on
3 filetype plugin indent on    "plugin , indent 한 번에 적용
4
5 set t_Co=256
6 set background=dark
7 let g:solarized_termcolors=16
8 colorscheme solarized
9
10 set nocompatible
11 set hlsearch                "highlight searches
12 set number                  "show line numbers
13 set ruler                    "show the current row and column
14
15 set tabstop=2               "Tab == 2 space
16 set shiftwidth=2            "auto indent 2 space
17
18 "set cindent "auto indent like C
19 set autoindent
20
21 set ls=2
22
23 "Beomjun Shin's vimrc
24 "below is temporary status line
25 set statusline=
26 set statusline +=%1*\ %n\ %*           "buffer number
27 set statusline +=%5*%{&ff}%*          "file format
28 set statusline +=%3*%y%*               "file type
```

여러분의 vim을 만드세요!

`syntax on` “문법강조

`colorscheme elflord` “vim 색깔테마 설정

`set nu` “set number을 축약해서 nu로 써도 됩니다.

`set ruler` “오른쪽 하단에 커서의 위치를 보여줌

`set number` “왼쪽에 행번호 표시

`set tabstop=2` “tab크기를 space 2개로

`set shiftwidth=2` “autoindent할 때 tab크기를 space 2개로

`set paste` “붙여넣기할 때 계단현상 방지

`set cindent` “자동 들여쓰기

우선 위의 설정만 다같이 해봅시다!

오늘의 과제!

Vim Setting을 다 한 뒤에 아래 코드를 넣고 저장한 뒤에
실행(웹브라우저 이용)한 화면을 검사받고 가세요 ~

```
<!DOCTYPE html> <!--ie구버전에서 안보일 때 지우기 -->  
<meta charset="utf-8">
```

```
<html>  
<head>  
  <title>Hello World!</title>  
</head>  
<body>  
  <h1> Hello Lion! </h1>  
  <p> let's study! </p> <br>  
  <!--100줄 이상 채우기-->  
</body>  
</html>
```

참고자료

<http://rubymonk.com>

<http://www.codecademy.com>

<http://www.code.org>

다음시간까지 과제

<http://www.codecademy.com/tracks/web>

<WebFundamental Courses>

1. Introduction to HTML
2. HTML Structure : Using Lists
3. HTML Structure: Tables. Divs and Spans
4. Introductino to CSS

위의 코스들을 전부 해보기

코스내에서 project들은 vim에서 똑같이 해보면서
vim 또한 숙달시켜웁니다! (복사,붙여넣기,Undo 등)

모르는거 질문은?

코스내에서 project들은 vim에서 똑같이 해보면서
vim 또한 숙달시켜웁니다! (복사,붙여넣기,Undo 등)

참고자료 - vim 책

