

전역

- :help keyword - keyword도움말
- :saveas file - file로 저장하기
- :close - 현재 영역 닫기
- K 커서위치한 단어에 대한맨페이지 열기
- 커서 이동
- H - 화면 상단으로 이동
- M - 화면 중간으로 이동
- L - 화면 하단으로 이동
- w - 다음 단어 시작으로 점프
- W - 다음 단어 시작으로 점프 (단어 옆에 붙은 연속된 문자부호도하나의 단어로본다)
- e - 다음 단어 끝으로 점프
- E - 다음 단어 끝으로 점프(단어 옆에 붙은 연속된 문자부호도 하나의 단어로 본다)
- b - 이전 단어 시작으로 점프
- B - 이전 단어 시작으로 점프 (단어 옆에 붙은 연속된 문자부호도 하나의 단어로봄)
- % - 현재괄호와 맞는 짝의괄호로 이동 (:h matchpairs를 통해 더많은 정보를얻음)
- O - 행의 처음으로 점프
- ^ - 행의 시작 문자로 점프
- \$ - 행의 끝으로 점프
- g_ - 행의 끝 문자로 점프
- gg - 문서 첫 행으로 점프
- G - 문서 마지막 행으로 점프
- 5G - 5번째 행으로 점프
- fx - 커서 위치 기준 오른쪽에서 가장 가까운 문자 x로 이동
- tx - 커서 위치 기준 오른쪽에서 가장 가까운 문자 x의 한 칸 뒤(왼쪽)으로 이동
- Fx - 커서 위치 기준 왼쪽에서 가장 가까운 문자 x로 이동
- Tx - 커서 위치 기준 왼쪽에서 가장 가까운 문자 x의 한 칸 앞(오른쪽)으로 이동
- : - 가장 최근의 f, t, F나 T 명령을 실행
- , - 가장 최근의 f, t, F나 T 명령을 반대 방향으로 다시 실행
- } - 다음 단락(또는 함수/블록)으로 이동
- { - 이전 단락(또는 함수/블록)으로 이동
- zz - 커서가 있는 행을 중간으로 하도록 화면이동
- u - 실행 취소
- Ctrl + r - 다시 실행

- Ctrl + e -커서이동없이한줄아래화면이동
- Ctrl + y -커서이동없이한줄위로화면이동
- Ctrl + b - 한 화면 위로
- Ctrl + f - 한 화면 아래로
- Ctrl + d - 반 화면 아래로
- Ctrl + u - 반 화면 위로

삽입 모드 - 텍스트 삽입/추가

- i - 커서 앞에 삽입
- I - 행 시작에 삽입
- a - 커서 뒤에 추가
- A - 행 끝에 추가
- o - 현재 행 아래에 새 행 추가
- O - 현재 행 위에 새 행 추가
- ea - 단어 끝에 추가
- Esc - 삽입 모드 종료

편집

- r - 한 글자 바꾸기
- J - 현재 행과 다음 행을 연결
- gj - 현재 행과 다음 행을 둘 사이에 공백 없이 연결
- gwip - 단락을 리플로우
- cc - 행 전체를 새로 쓰기
- C - 행 끝까지를 새로 쓰기
- c\$ - 행 끝까지를 새로 쓰기
- ciw - 단어 전체를 새로 쓰기
- cw - 단어 끝까지를 새로 쓰기
- s - 한 글자 삭제하고 삽입 모드 시작
- S -행 삭제하고 텍스트 입력 (<kbd>cc</kbd>와 동일)
- xp - 두 문자 위치 바꾸기 (잘라내기 붙여넣기 조합)
- . - 마지막 명령 반복
- 텍스트 선택 (비주얼 모드)
- v - 선택 모드 시작. 텍스트 선택해서 명령 수행 (가령 y로 복사)
- V - 행 단위 선택 모드 시작
- o - 선택 영역의 반대쪽 끝으로 이동
- Ctrl + v - 블록 선택 모드 시작
- O - 블록의 반대쪽 모서리로 이동
- aw - 단어 선택
- ab - () 블럭 선택
- aB - {} 블럭 선택
- ib - () 블럭의 내부 선택
- iB - {} 블럭의 내부 선택

- Esc - 선택 모드 종료

선택 모드 명령

- > - 텍스트를 오른쪽으로 이동
- < - 텍스트를 왼쪽으로 이동
- y - 선택한 텍스트 복사
- d - 선택한 텍스트 삭제
- ~ - 대소문자 반전

레지스터

- :reg - 레지스터 내용물 표시
- "xy - 레지스터 x로 복사
- "xp - 레지스터 x의 내용물 붙여넣기

Tip 레지스터들은 ./viminfo에 저장되며 다음번 vim 재시작 때 다시 읽어들임.

Tip 0번 레지스터에는 항상 최근 복사 명령의 값이 들어있음.

표시

- :marks - 표시 항목 표시
- ma - 현재 위치를 표시 a로 설정
- `a - 표시 a의 위치로 점프
- y`a - 표시 a의 위치까지 복사

매크로

- qa - 매크로 a 기록 시작
- q - 매크로 기록 중지
- @a - 매크로 a 실행
- @@ - 마지막 실행한 매크로 재실행

잘라내기와 붙여넣기

- yy - 행 복사
- 2yy - 2줄 복사
- yw - 커서 위치에서 다음 단어 시작까지 복사
- y\$ - 행 끝까지 복사
- p - 커서 위치 뒤에 붙여넣기
- P - 커서 위치 앞에 붙여넣기
- dd - 행 잘라내기
- 2dd - 2줄 잘라내기
- dw - 커서 위치에서 다음 단어 시작까지 잘라내기
- D - 행 끝까지 잘라내기
- d\$ - 행 끝까지 잘라내기
- x - 한 글자 잘라내기

나가기

- :w - 저장하고, 나가지는 않기
- :w !sudo tee % - sudo로현재파일 저장
- :wq or :x or ZZ - 저장하고 나가기
- :q - 나가기 (저장하지 않은 변경 사항 있으면 실패)
- :q! or ZQ - 나가기 (저장하지 않은 변경 사항 버림)
- :wqa - 열어 놓았던 모든 탭을 다 저장하고 나가기

찾기와 바꾸기

- /pattern - 패턴 찾기
- ?pattern - 패턴 역방향 찾기
- \vpattern - '마법' 패턴: 영문/숫자가 아닌 문자는 정규표현식 심볼로 해석 (이스케이프 불필요)
- n - 같은 방향으로 찾기 반복
- N - 반대 방향으로 찾기 반복
- :%s/old/new/g - 파일 내 모든 old를 new로 바꾸기
- :%s/old/new/gc - 파일 내 모든 old를 new로 확인하며 바꾸기
- :noh - 찾기 강조 표시 없애기
- 여러 파일에서 찾기
- :vimgrep /pattern/ {`file`} - 여러 파일에서 패턴 찾기
- e.g. :vimgrep /foo/ **/*
- :cn - 다음 일치 항목으로 점프
- :cp - 이전 일치 항목으로 점프
- :copen - 일치 목록을 새 창으로 열기

여러 파일 작업

- :e file - 새 버퍼에서 파일 편집
- :bnext or :bn - 다음 버퍼로 이동
- :bprev or :bp - 이전 버퍼로 이동
- :bd - 버퍼 삭제 (파일 닫기)
- :ls - 열린 버퍼 모두 나열
- :sp file - 새 버퍼에 파일 열고 상하로 창 분할
- :vsp file - 새 버퍼에 파일 열고 좌우로 창분할
- Ctrl + ws - 상하로 창 분할
- Ctrl + ww - 창 전환
- Ctrl + wq - 창 닫기

- Ctrl + ww - 좌우로 창 분할
- Ctrl + wh - 왼쪽 창으로이동(좌우 분할)
- Ctrl + wl - 오른쪽창으로 이동 (좌우 분할)
- Ctrl + wj - 아래 창으로 이동(상하 분할)
- Ctrl + wk - 위 창으로 이동 (상하 분할)

탭

- :tabnew or :tabnew {page.words.file} - 새 탭에서 파일 열기
- Ctrl + wT - 현재분할창을 새로운 탭으로이동
- gt or :tabnext or :tabn -다음 탭 이동
- gT or :tabprev or :tabp - 이전 탭이동
- #gt - #번 탭으로 이동
- :tabmove # - 현재 탭을 #번째로 이동 0부터
- :tabclose or :tabc - 현재 탭과 그 안의 창들 닫기
- :tabonly or :tabo - 현재탭 이외 모든 탭 닫기
- :tabdo command - 모든 탭에서 command 실행(예를 들어, :tabdo q - 열린 모든 탭 닫기)
- G 파일 끝으로 가기
- gg 파일 처음으로 가기
- 10G 처음에서 10번줄로 가기
- CTRL+g show file information
- n goto next
- N goto previous
- :s/find/found find를 found로 바꾸기
- :s/asd/qwe/g 줄에 있는 모든 asd -> qwe
- :%s/asd/qwe 모든줄에 있는 첫번째 asd -> qwe
- :5,10s/asd/qwe 5-10줄에있는 첫번째 asd -> qwe
- :!<command> bash에서 커맨드 실행
- 다른이름으로 저장해도 현재파일 편집 그대로 유지
- :q 나가기, :q! 저장안하고 나가기
- 선택한 상태에서 커맨드 하면 :`<`,`>생김 -> 구간에 대해 커맨드
- :r filename 파일에서 읽어오기
- :linenumber linenumber로 이동