

기본 터미널 Zsh 적용

zsh 란?

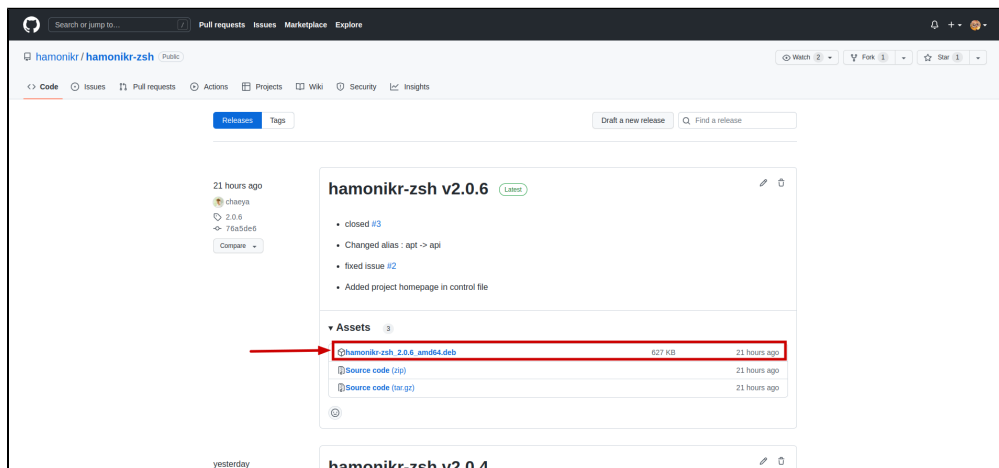
zsh(z shell)는 하모니카 사용자에게 터미널 환경의 사용성을 개선해주는 쉘 스크립트입니다.

zsh의 더 자세한 설명은 <https://github.com/hamonikr/hamonikr-zsh> 에서 확인할 수 있습니다.

zsh 설치

하모니카 6.0에는 기본적으로 zsh가 설치되어 있지만 zsh를 새로 설치해야 하는 경우엔 아래와 같은 방법으로 설치가 가능합니다.

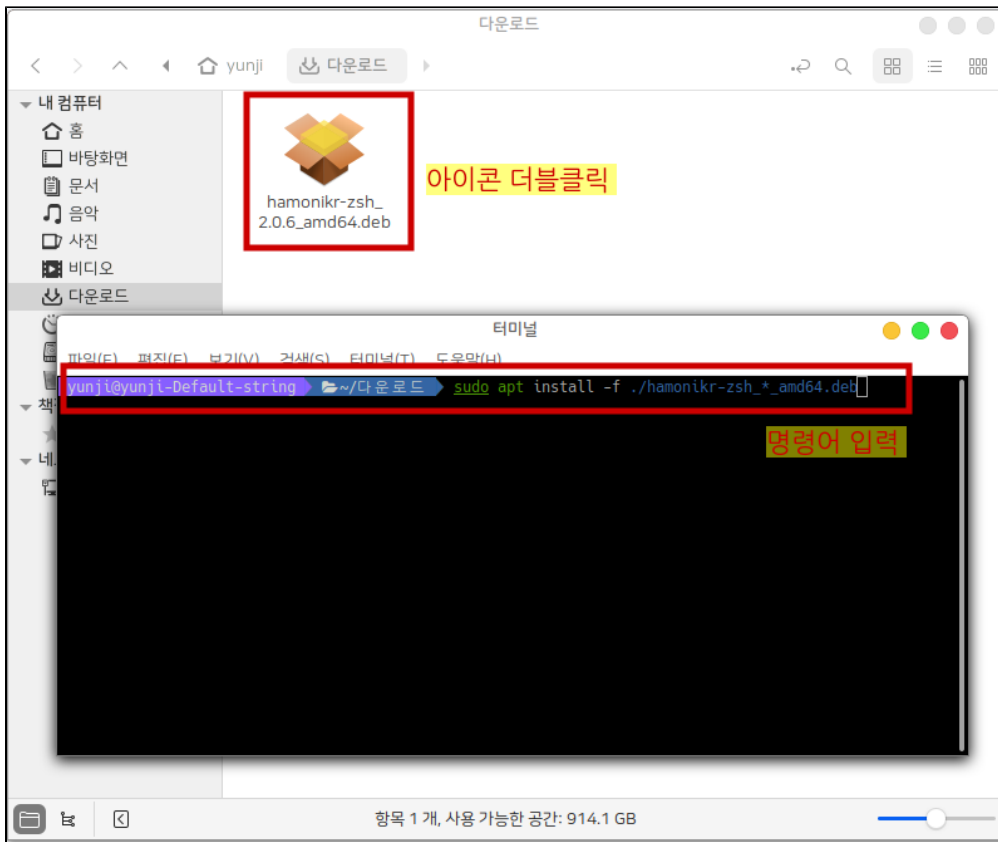
최신 패키지 다운로드



아래의 링크를 통해 최신 패키지를 다운로드 받아줍니다.

<https://github.com/hamonikr/hamonikr-zsh/releases>

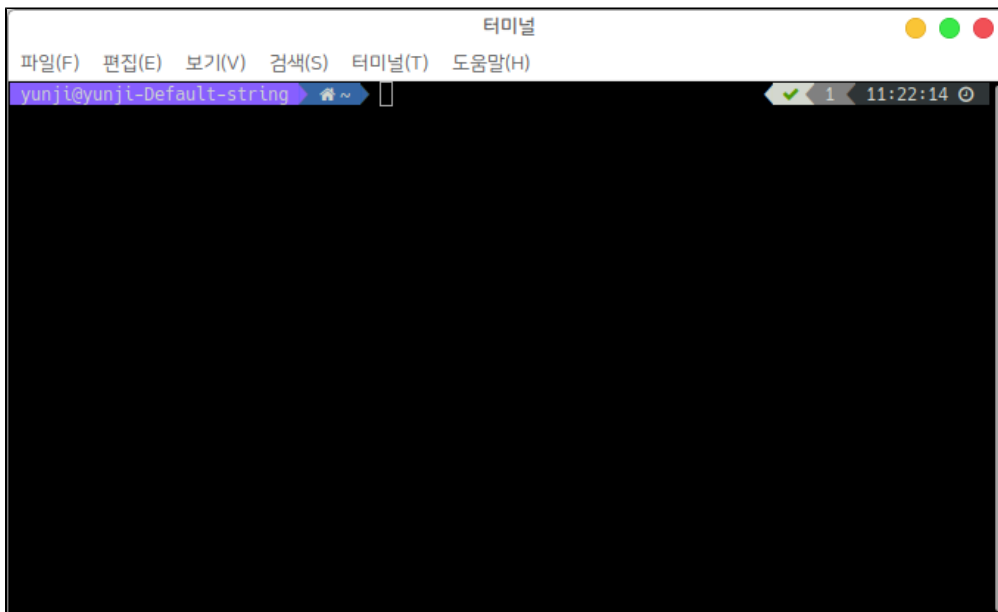
패키지 설치



다운로드 받은 파일이 있는 폴더를 열어줍니다.

우클릭 후 이 위치로 터미널 열기 메뉴를 누르고 `sudo apt install -f ./hamonikr-zsh*_amd64.deb` 을 입력하거나 패키지 아이콘을 더블클릭하면 설치가 진행됩니다.

(※ 패키지 설치 후 PC 재시작을 해야 새로운 환경이 적용됩니다.)

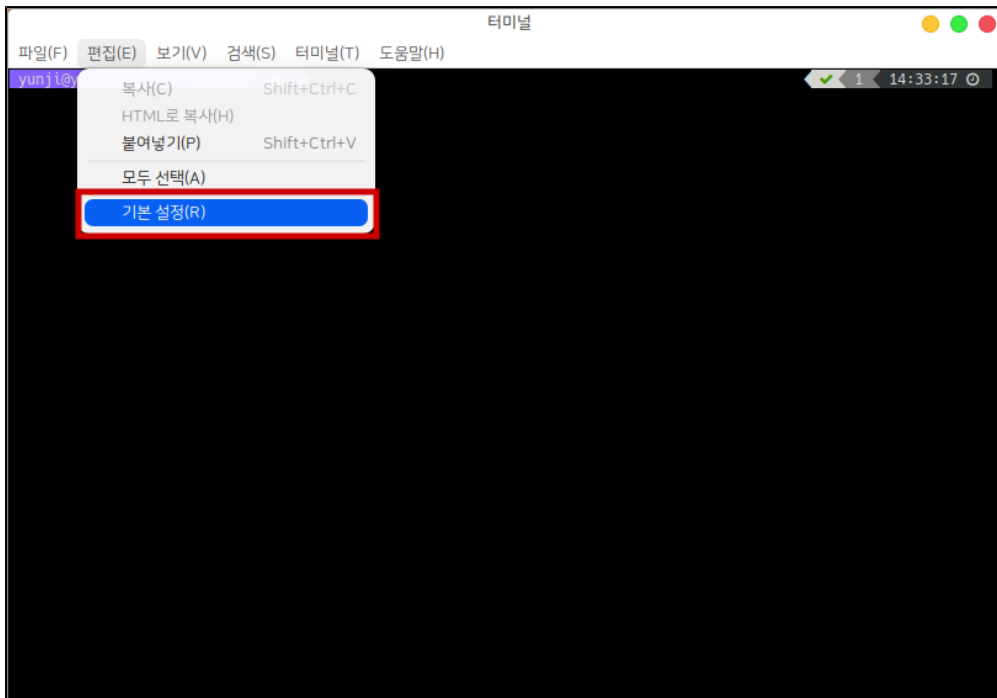


정상적으로 설치가 완료되면 터미널을 열었을 때 아래와 같은 화면이 보여집니다.

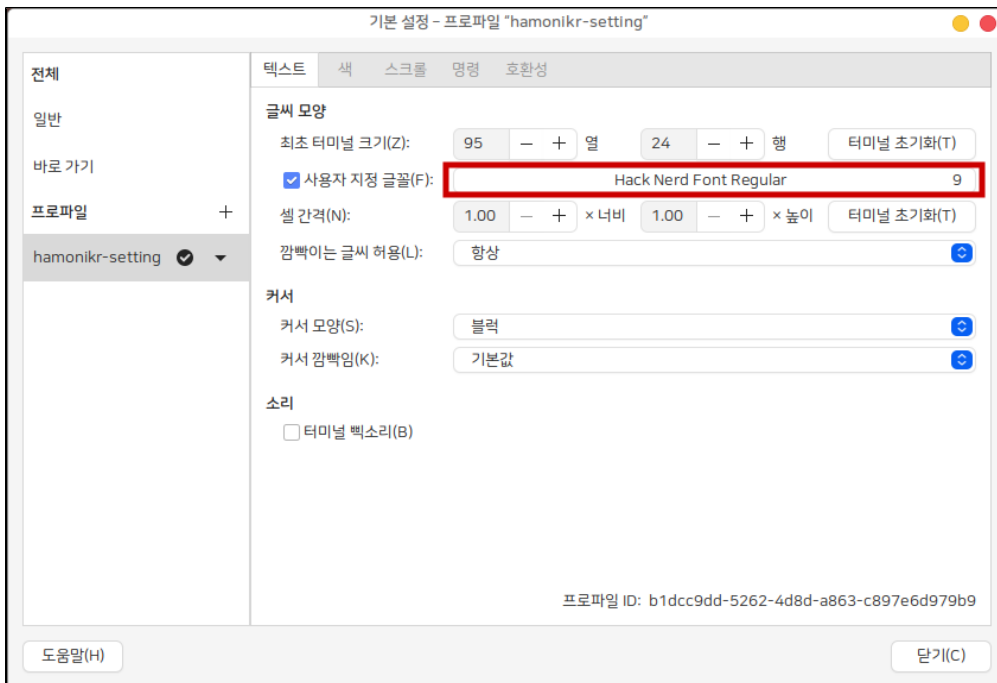
폰트설정 변경

zsh를 열었을 때 터미널 폰트가 적용되지 않은 경우 폰트설정을 변경하는 것이 가능합니다.

터미널에서 폰트설정



터미널을 열고 편집 하위메뉴의 기본설정 버튼을 눌러 기본설정창을 열어줍니다.

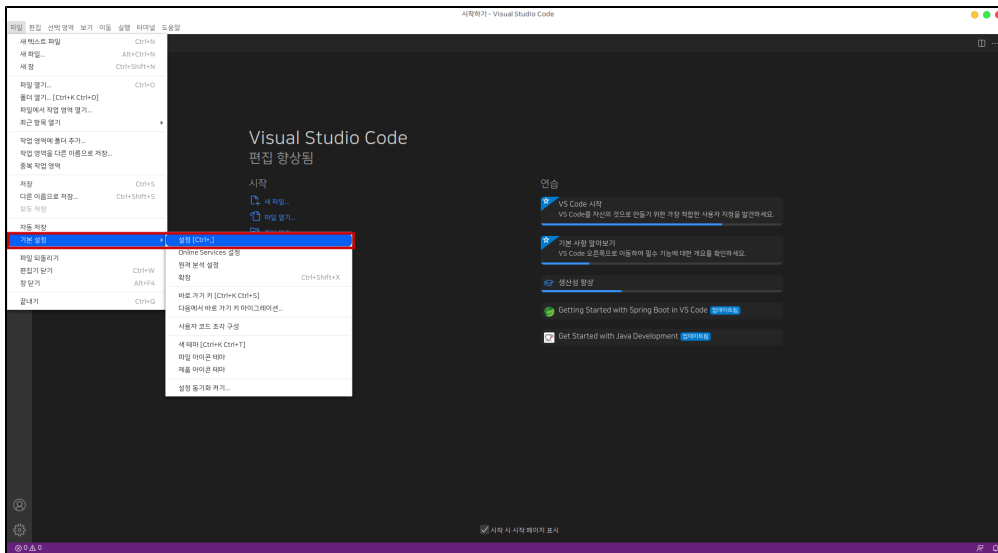




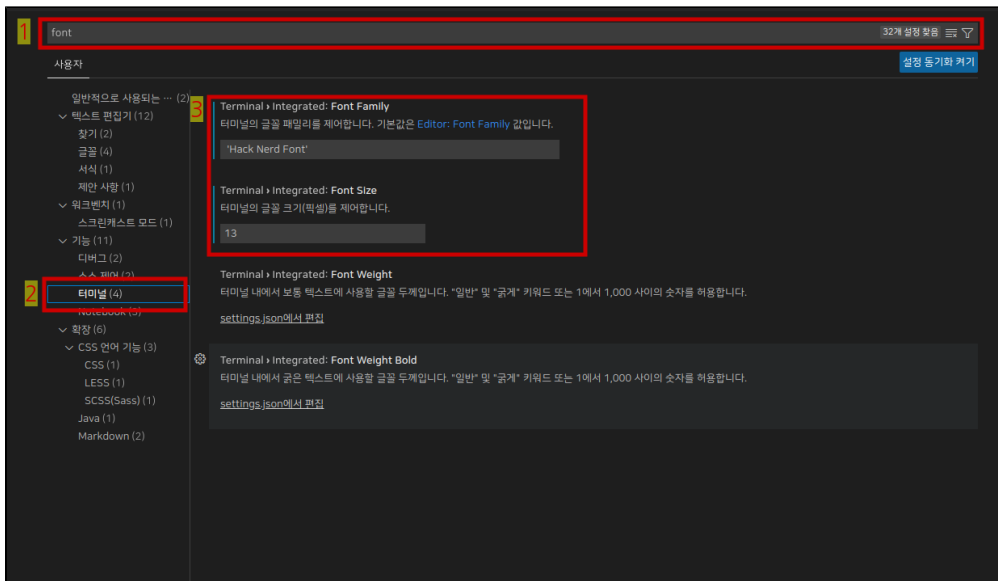
기본설정창 좌측 메뉴의 hamonikr-settings를 클릭한 후 사용자 지정 글꼴을 클릭하면 터미널 글꼴 선택 팝업화면이 보여집니다.

팝업화면의 검색창에서 **Hack Nerd Font**를 검색 후 하단의 폰트 크기를 9로 맞춰준 뒤 선택 버튼을 누르면 설정이 완료됩니다.

vscode에서 폰트설정



vscode에서 파일 하위메뉴의 기본설정 메뉴를 눌러 설정화면에 접속합니다.



설정 화면 상단에 **font**를 검색하고 좌측 메뉴의 터미널 메뉴를 클릭하면 터미널 폰트설정을 할 수 있는 화면이 보여집니다.

입력란에 **Hack Nerd Font** 을 입력 후 하단의 폰트사이즈를 13으로 설정하면 폰트설정이 완료됩니다.

zsh의 기능 및 사용법

하모니카에 설치된 zsh의 주요기능들을 설명합니다.

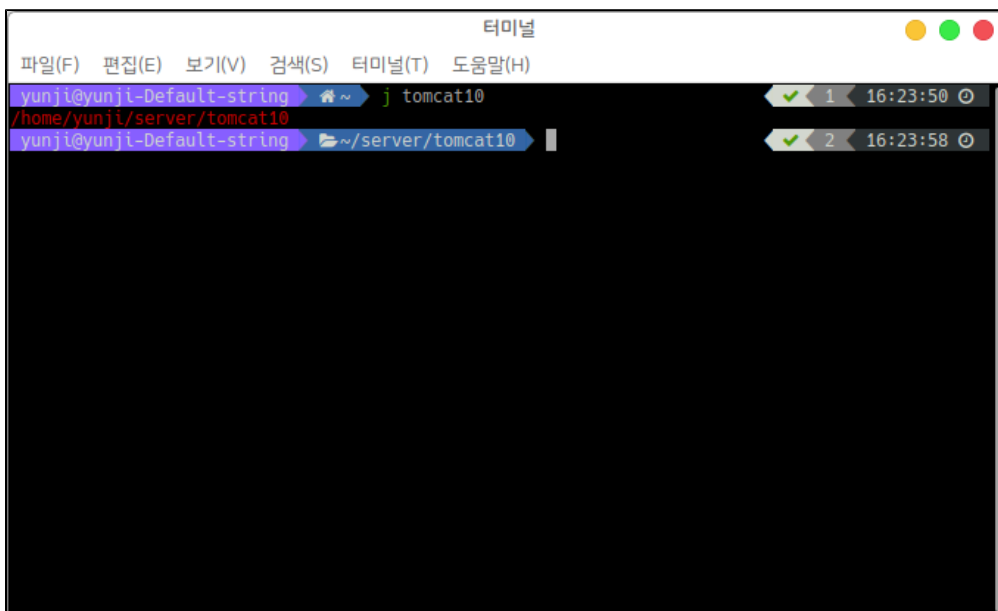
autojump

autojump는 여러번의 cd 명령을 통해서 디렉터리를 바꾸는 것을 간단히 j 명령을 통해서 한번에 이동 가능하게 하는 기능입니다.

autojump 옵션은 **autojump --help**에서 확인이 가능합니다.

(* j 명령어는 명령어를 사용하기 이전에 이동한적이 있었던 디렉토리에 한해서 사용 가능)

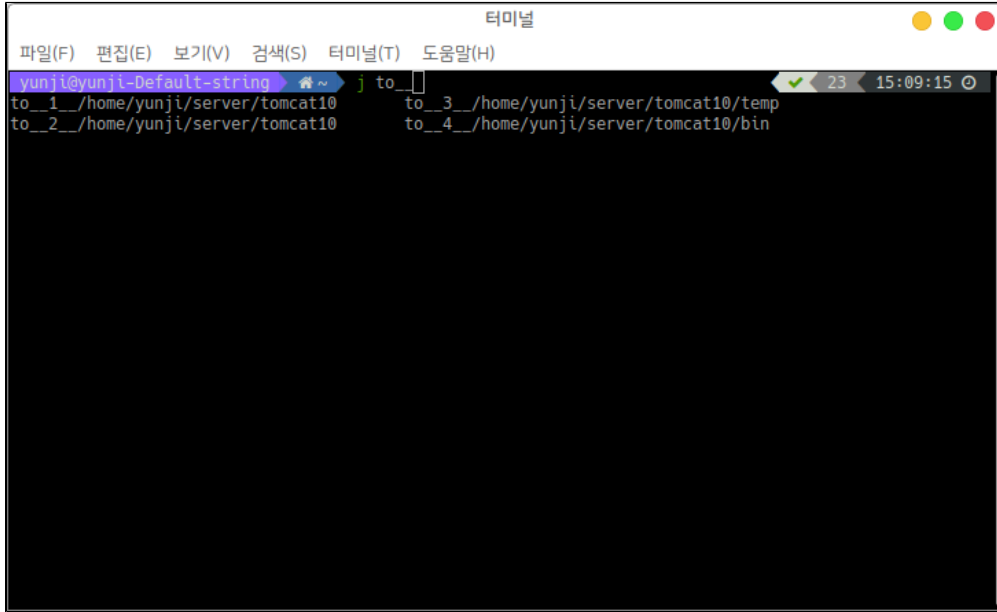
j 명령어



예를들어 'tomcat10' 을 포함하는 디렉토리로 이동하고 싶은 경우: 터미널에 **j tomcat10** 입력

이동 시에 파일명을 끝까지 입력하지 않아도(j to 혹은 j t) 겹치는 파일 이름이 존재하지 않으면 이동이 가능합니다.

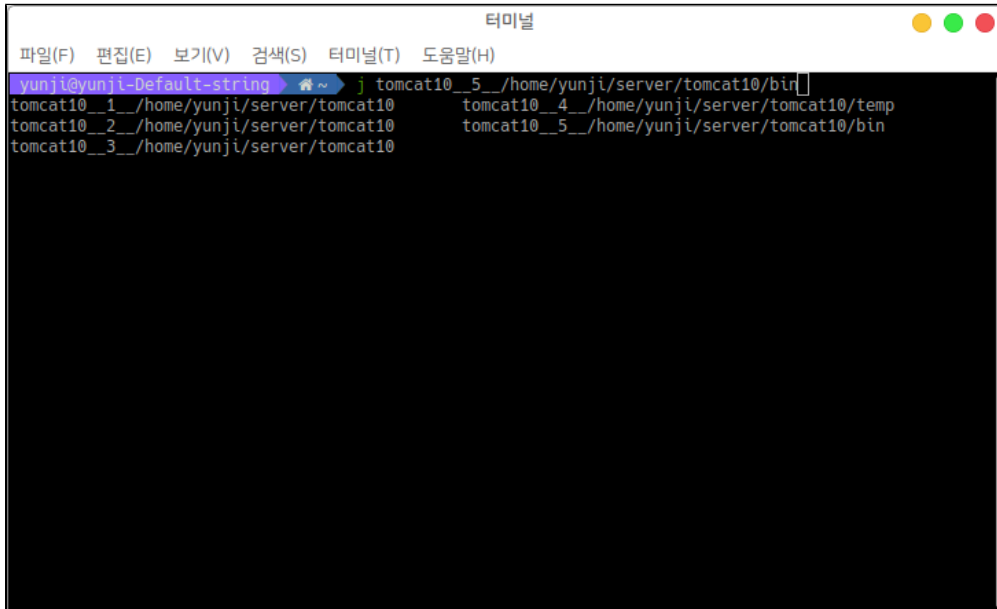
(파일이 겹칠시에는 우선순위가 높은 곳으로 이동 - 우선순위 설정 방법은 3.1.4에서 설명)



j 명령어를 사용할 때 이동할 디렉토리의 전체 경로를 보고 싶으면 j 파일명을(앞부분만 입력해도 확인 가능) 입력하고 **tab** 키를 두번 눌러줍니다.

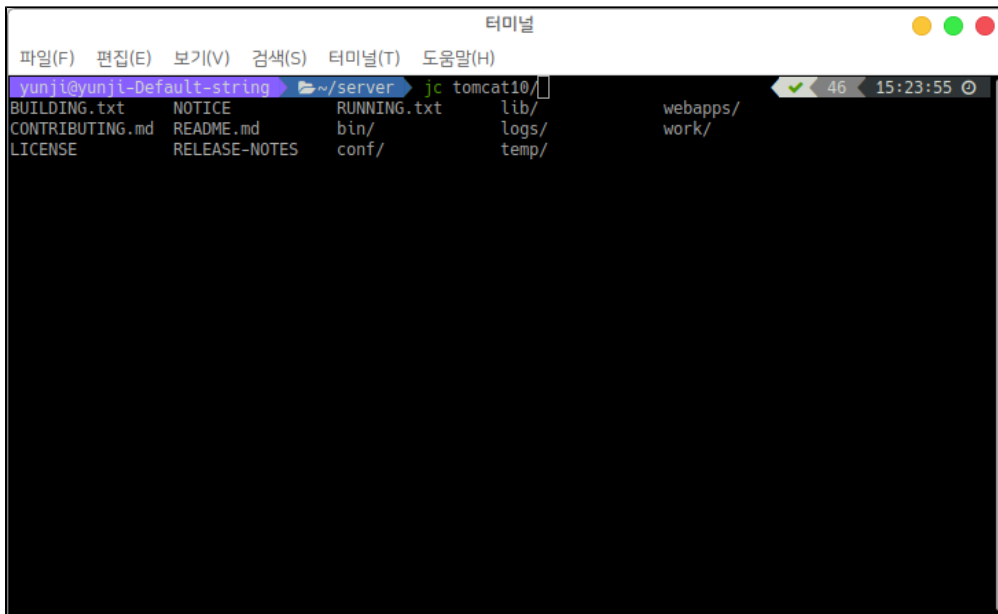
j tomcat10을 입력 후 tab키를 누르면 아래와 같이 경로 목록이 나오고, 목록이 나온 후 tab 키를 더 누르면 아래의 경로들로 이동이 가능합니다.

<tab을 두 번 눌러서 경로가 나온 화면>



<tab을 더 누르면 아래의 경로들로 이동이 가능함>

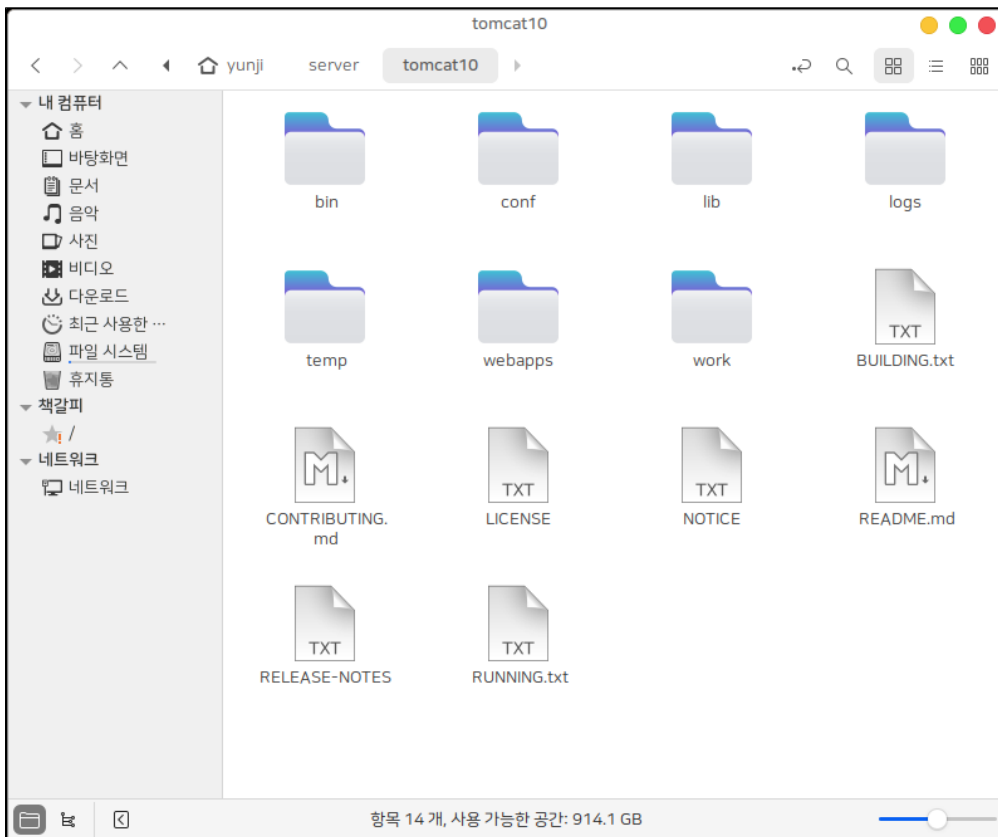
jc 명령어



jc 명령어는 디렉토리 이동 기능은 j와 같으나 **tab**키를 누르면 전체경로가 아닌 해당 디렉토리의 하위 디렉토리와 파일을 보여줍니다.

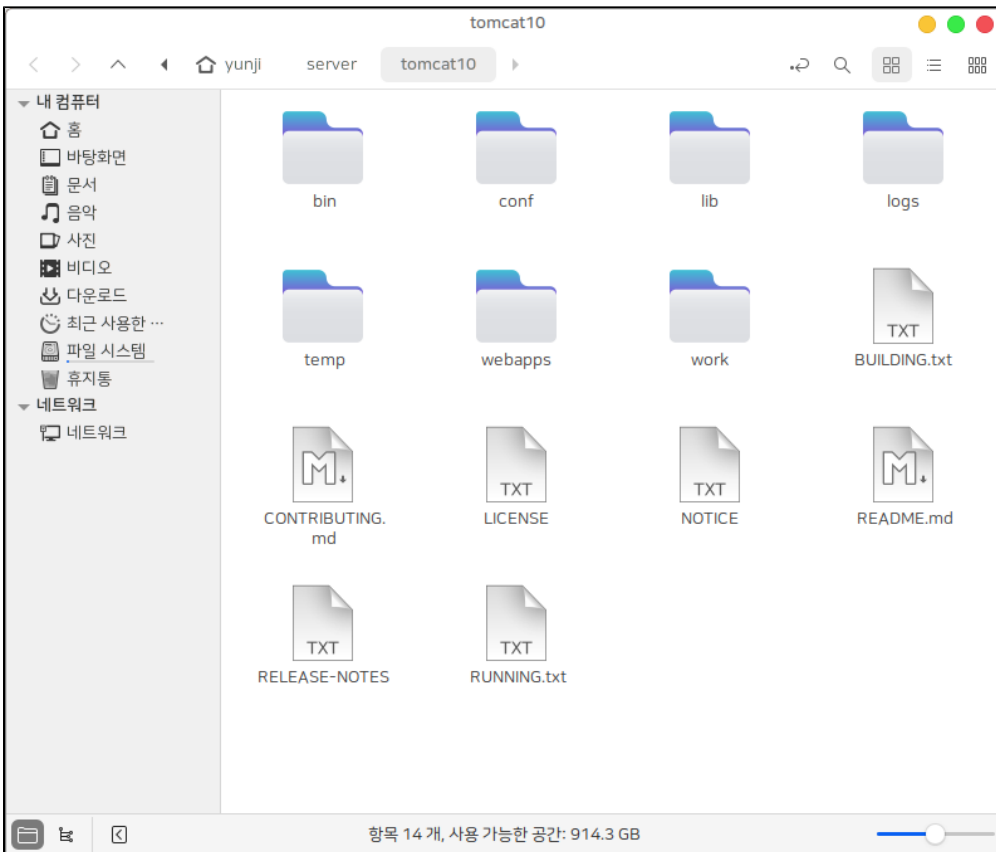
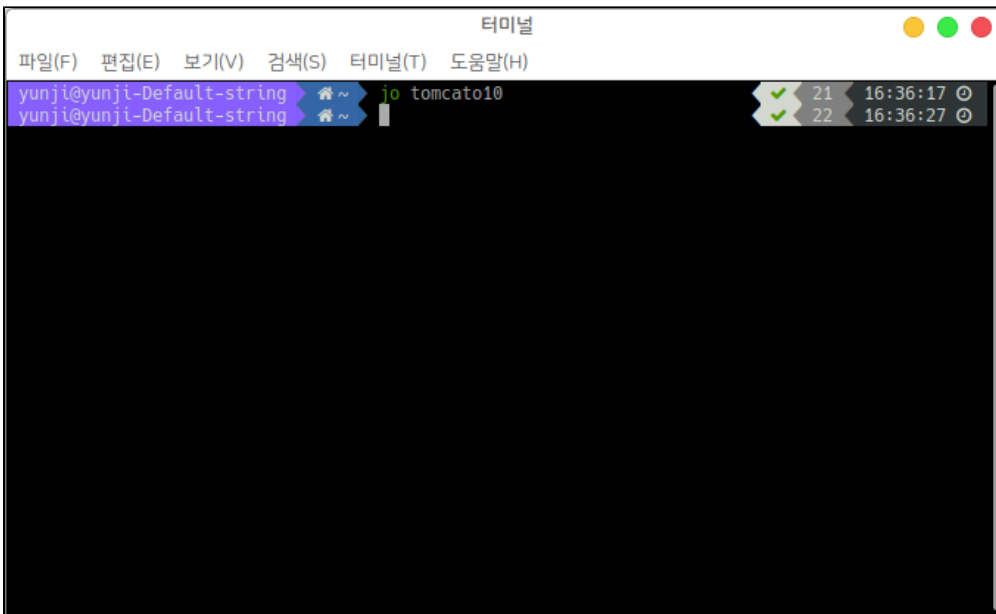
jc 명령어를 이용해 하위 디렉토리를 확인하려면 상위 디렉토리에서 명령어를 입력해야 합니다.

<jc를 이용해 tomcat10의 하위 디렉토리와 파일 확인하는 화면>



<jc를 이용해 확인한 하위 디렉토리 목록과 동일함>

jo 명령어



jo 명령어는 디렉토리로 이동하는 대신 파일 탐색기 창을 해당 디렉토리로 열 수 있습니다.

tomcat10 디렉토리를 열기 위해 **jo tomcat10** 명령어 입력하면 파일 탐색기가 보여집니다.

<jo를 사용해 tomcat10 디렉토리를 여는 과정>

파일명이 중복될 때 우선순위 설정



```
yunji@yunji-Default-string ~$ autojump --i
44.4: /home/yunji/server/tomcat10
yunji@yunji-Default-string ~$ autojump --d
29.4: /home/yunji/server/tomcat10
yunji@yunji-Default-string ~$ autojump
44.4: /home/yunji/server/tomcat10
```

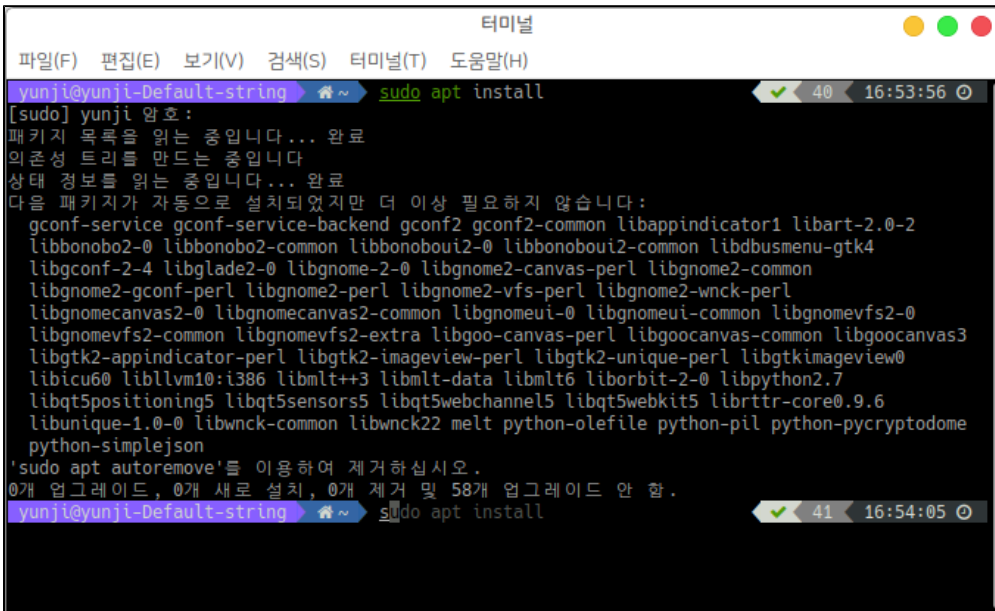
j 명령어를 사용할 때 이름이 동일한 디렉토리나 파일이 있으면 우선순위가 높은 곳으로 이동합니다.

우선순위를 높이고 싶을 때는 해당 디렉토리로 이동해서 `autojump --i` 명령어를 입력하고, 낮추고 싶을 때는 `autojump --d` 명령어를 입력합니다.

명령어를 입력하면 변경된 우선순위가 보여집니다.

<tomcat10 디렉토리의 우선순위를 변경하는 화면>

autosuggestion

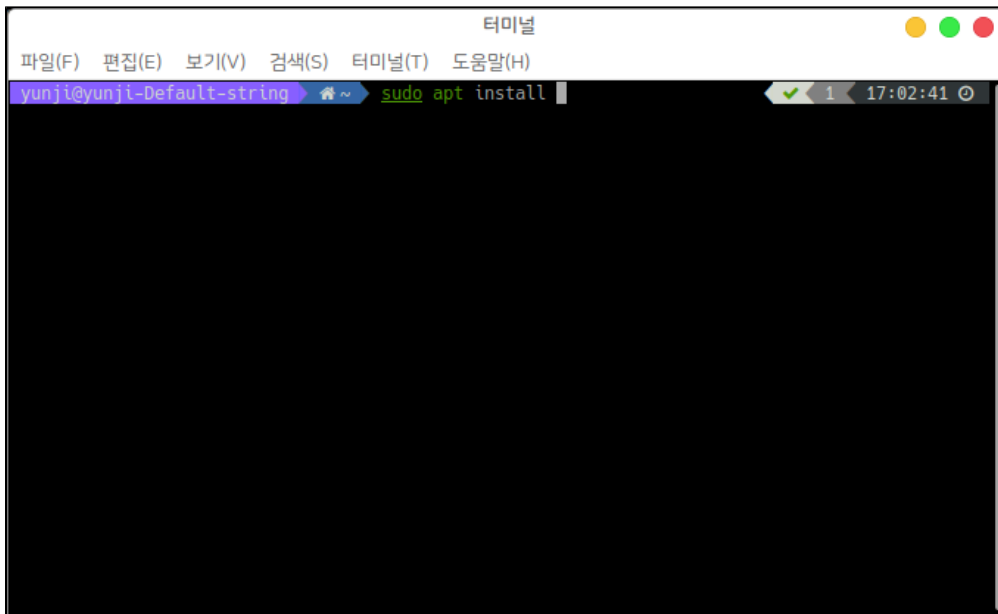


```
yunji@yunji-Default-string ~$ sudo apt install
[sudo] yunji 암호:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다:
gconf-service gconf-service-backend gconf2 gconf2-common libappindicator1 libart-2.0-2
libbonobo2-0 libbonobo2-common libbonoboui2-0 libbonoboui2-common libdbusmenu-gtk4
libgconf-2-4 libglade2-0 libgnome-2-0 libgnome2-canvas-perl libgnome2-common
libgnome2-gconf-perl libgnome2-perl libgnome2-vfs-perl libgnome2-wnck-perl
libgnomecanvas2-0 libgnomecanvas2-common libgnomeui-0 libgnomeui-common libgnomevfs2-0
libgnomevfs2-common libgnomevfs2-extra libgoo-canvas-perl libgoocanvas-common libgoocanvas3
libgtk2-appindicator-perl libgtk2-imageview-perl libgtk2-unique-perl libgtkimageview0
libicu60 libllvm10:i386 libmlt++3 libmlt-data libmlt6 liborbit-2-0 libpython2.7
libqt5positioning5 libqt5sensors5 libqt5webchannel5 libqt5webkit5 librttr-core0.9.6
libunique-1.0-0 libwnck-common libwnck22 melt python-olefile python-pil python-pycryptodome
python-simplejson
'sudo apt autoremove'를 이용하여 제거하십시오.
0개 업그레이드, 0개 새로 설치, 0개 제거 및 58개 업그레이드 안 함.
yunji@yunji-Default-string ~$ sudo apt install
```

이전에 입력한 적이 있는 명령어는 명령을 다 입력하기 전에 옅은 회색으로 완성된 내용이 표시됩니다.

옅은 회색으로 나온 명령어를 완성시키려면 키보드 방향키 `"→"`를 눌러줍니다.

syntax-highlighting



명령어 오류를 잡을 때 명령을 실행하기 전에 검토하는 데 도움이 되는 명령어 강조 표시 기능입니다.

맞는 명령어는 초록색으로 표시되고, 잘못된 명령어는 빨간색으로 표시됩니다.

<맞는 명령어>



<잘못된 명령어>

Useful Alias

자주 사용하는 명령어에 단축어를 사용해서 시간과 노력을 절약할 수 있는 기능입니다.

단축어기능을 사용하면 사용자가 터미널 사용을 간단하게 개인화할 수 있습니다.

모든 단축어 표시

```
터미널
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
yunji@yunji-Default-string ~$ alias
alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "${history|t
ail -n1|sed -e '\s/^s*[0-9]\+s*//;s/[:&]\s*alert$/\s*'\''"'
apt='sudo apt'
apu='sudo apt update'
code='code --no-sandbox'
count='find . -type f | wc -l'
cpv='rsync -ah --info=progress2'
df='df -h'
egrep='egrep --color=auto'
fgrep='fgrep --color=auto'
grep='grep --color=auto'
l='ls -CF'
la='ls -A'
left='ls -t -1'
ll='ls -aF'
ls='ls --color=auto'
lt='ls --human-readable --size -1 -S --classify'
myip='curl http://ipecho.net/plain; echo'
which-command=whence
yunji@yunji-Default-string ~$
```

터미널에 **alias** 를 입력하면 모든 단축어 확인이 가능합니다.

단축어 만들기

```
터미널
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
zsh: command not found: command_to_perform
yunji@yunji-Default-string ~$ alias ctp ="command to perform"
zsh: command to perform not found
yunji@yunji-Default-string ~$ alias ctp ="command_to_perform"
zsh: command_to_perform not found
yunji@yunji-Default-string ~$ alias ctp="command_to_perform"
yunji@yunji-Default-string ~$ alias
alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "${history|t
ail -n1|sed -e '\s/^s*[0-9]\+s*//;s/[:&]\s*alert$/\s*'\''"'
alias_name=command_to_perform
apt='sudo apt'
apu='sudo apt update'
code='code --no-sandbox'
count='find . -type f | wc -l'
cpv='rsync -ah --info=progress2'
ctp=command_to_perform
df='df -h'
egrep='egrep --color=auto'
fgrep='fgrep --color=auto'
grep='grep --color=auto'
l='ls -CF'
la='ls -A'
left='ls -t -1'
ll='ls -aF'
ls='ls --color=auto'
lt='ls --human-readable --size -1 -S --classify'
myip='curl http://ipecho.net/plain; echo'
which-command=whence
yunji@yunji-Default-string ~$
```

alias = " " 터미널에 입력하면 단축어 생성이 가능합니다. (단축어를 설정할 때 명령은 등호와 명령 사이에 공백 없이 따옴표로 묶어야 합니다.)

단축어 생성 후 **alias** 명령어를 이용해 생성한 단축어를 확인할 수 있습니다.

(※ 이러한 방법으로 생성한 단축어는 PC 재시작 시 사라집니다.)

<단축어를 만들고 **alias**로 단축어 목록을 확인하면 생성된 단축어 확인 가능>

영구적인 단축어 만들기

```
drwxrwxr-x 4 yunji yunji 4096 9월 22 11:40 .npm
drwx----- 3 yunji yunji 4096 7월 29 11:44 .pki
-rw-r--r-- 1 yunji yunji 807 7월 29 11:37 .profile
-rw----- 1 yunji yunji 80 8월 22 13:48 .psql_history
drwx----- 2 yunji yunji 4096 8월 24 14:26 .putty
drwxrwxr-x 3 yunji yunji 4096 9월 22 17:18 .shutter
drwx----- 2 yunji yunji 4096 8월 24 14:33 .ssh
drwxrwxr-x 3 yunji yunji 4096 8월 12 13:49 .sts4
-rw-r--r-- 1 yunji yunji 0 7월 29 11:58 .sudo_as_admin_successful
drwxrwxr-x 2 yunji yunji 4096 7월 29 16:25 .swt
drwx----- 3 yunji yunji 4096 9월 5 16:09 .thumbnails
-rw----- 1 yunji yunji 13563 9월 28 17:58 .viminfo
drwxrwxr-x 3 yunji yunji 4096 7월 29 13:19 .vscode
-rw-rw-r-- 1 yunji yunji 247 9월 16 13:05 .wget-hsts
drwxrwxr-x 4 yunji yunji 4096 9월 28 14:08 .wine
-rw-rw-r-- 1 yunji yunji 131 7월 29 02:40 .xinputrc
-rw----- 1 yunji yunji 12881 9월 28 18:03 .xsession-errors
-rw----- 1 yunji yunji 18138 9월 28 17:59 .xsession-errors.old
-rw-rw-r-- 1 yunji yunji 49023 9월 28 11:03 .zcompdump
-rw-r--r-- 1 yunji yunji 1540 9월 28 17:58 .zshrc
-rw-rw-r-- 1 yunji yunji 4096 9월 22 16:48 Downloads
-rw-rw-r-- 1 yunji yunji 38494576 1월 13 2016 SafariSetup.exe
drwxrwxr-x 3 yunji yunji 4096 8월 24 11:53 build
drwxrwxr-x 2 yunji yunji 4096 8월 24 11:53 cd
drwxrwxr-x 7 yunji yunji 4096 9월 8 11:09 hamonikr
drwxrwxr-x 3 yunji yunji 4096 8월 31 17:22 hamonikr_apps
drwxrwxr-x 3 yunji yunji 4096 8월 18 17:01 hamonize
drwxrwxr-x 7 yunji yunji 4096 9월 5 14:10 html-study
drwxrwxr-x 2 yunji yunji 4096 9월 2 18:08 key
drwxrwxr-x 3 yunji yunji 4096 8월 2 11:36 server
drwxrwxr-x 4 yunji yunji 4096 8월 12 16:13 springstudy
drwxrwxr-x 15 yunji yunji 4096 9월 28 14:33 translate
-rw-rw-r-- 1 yunji yunji 3220 12월 19 2018 winehq.key
drwxr-xr-x 2 yunji yunji 4096 7월 29 11:40 공개
drwxr-xr-x 4 yunji yunji 4096 9월 28 14:17 다운로드
drwxr-xr-x 3 yunji yunji 4096 9월 21 18:03 문서
drwxr-xr-x 6 yunji yunji 4096 9월 28 17:02 바탕화면
drwxr-xr-x 3 yunji yunji 4096 9월 16 13:11 비디오
drwxr-xr-x 2 yunji yunji 4096 9월 28 17:29 사진
drwxr-xr-x 2 yunji yunji 4096 7월 29 11:40 음악
drwxr-xr-x 2 yunji yunji 4096 7월 29 11:40 템플릿
yunji@yunji-Default-string ~ 26 18:04:55
```

위의 방법으로 생성된 단축어는 PC를 재시작하면 사라지기 때문에 영구적인 단축어를 생성하려면 아래와 같은 방법을 사용해야 합니다.

- 터미널에 **ls -al** 명령어를 입력 후 디렉토리에 있는 내용 리스트 중 **.zshrc** 파일을 열어줍니다. (.zshrc 파일을 여는 명령어 : **vi .zshrc**)

```
터미널
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

# Prompt settings
POWERLEVEL9K_LEFT_PROMPT_ELEMENTS=(context dir vcs)
POWERLEVEL9K_RIGHT_PROMPT_ELEMENTS=(status root_indicator background_jobs history time)
POWERLEVEL9K_MODE='nerdfont-complete'
POWERLEVEL9K_COLOR_SCHEME='light'
POWERLEVEL9K_CONTEXT_DEFAULT_BACKGROUND='slateblue1'
POWERLEVEL9K_CONTEXT_DEFAULT_FOREGROUND='white'

source /usr/share/powerlevel9k/powerlevel9k.zsh-theme
source /usr/share/autojump/autojump.sh
source /usr/share/zsh-autosuggestions/zsh-autosuggestions.zsh
source /usr/share/zsh-syntax-highlighting/zsh-syntax-highlighting.zsh
autoload -U +X compinit && compinit
autoload -U +X bashcompinit && bashcompinit
source /usr/share/bash-completion/bash_completion &>/dev/null

# dircolors
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# Alias
alias ll='ls -aF'
alias la='ls -A'
alias l='ls -CF'
alias alert='notify-send --urgency=low -i "${[ $? = 0 ] && echo terminal || echo error}" "${history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/;;s/[]&[]\s*alert$//'\`}"'
alias myip='curl http://ipecho.net/plain; echo'
alias apt='sudo apt'
alias apu='sudo apt update'
alias df='df -h'
alias lt='ls --human-readable --size -1 -S --classify'
alias left='ls -t -1'
alias count='find . -type f | wc -l'
alias cpv='rsync -ah --info=progress2'
alias code='code --no-sandbox'
alias hello='echo alias-test'

~
~
~
40,29 모두
```

b. vi .zshrc를 터미널에 입력하면 아래와 같은 화면이 보여지고 단축어 생성이 가능합니다.

1. i, a, o .
2. alias =" " .
3. esc :wq! .

```
yunji@yunji-Default-string ~$ hello
alias-test
yunji@yunji-Default-string ~$
```

생성한 단축어를 입력하면 정상적으로 단축어가 등록된 것을 확인할 수 있습니다.

단축어 삭제

```
터미널
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

yunji@yunji-Default-string ~$ unalias hello
yunji@yunji-Default-string ~$ alias
alert='notify-send --urgency=low -i "${[ $? = 0 ] && echo terminal || echo error}" "${history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/;;s/[]&[]\s*alert$//'\`}"'
apt='sudo apt'
apu='sudo apt update'
code='code --no-sandbox'
count='find . -type f | wc -l'
cpv='rsync -ah --info=progress2'
df='df -h'
egrep='egrep --color=auto'
fgrep='fgrep --color=auto'
grep='grep --color=auto'
l='ls -CF'
la='ls -A'
left='ls -t -1'
ll='ls -aF'
ls='ls --color=auto'
lt='ls --human-readable --size -1 -S --classify'
myip='curl http://ipecho.net/plain; echo'
which-command=whence
yunji@yunji-Default-string ~$
```

생성한 단축어는 **unlias 삭제할 단축어** 의 형태로 터미널에 입력하면 삭제가 가능합니다.