

Homebrew 빠른 참조

패키지 관리, Cask, Tap, 서비스, 정리

설치

Homebrew 설치

```
/bin/bash -c "$(curl -fsSL \
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.
sh)"
brew --version      # verify installation
```

설치 후 설정 (Apple Silicon)

```
# add Homebrew to PATH (Apple Silicon default)
echo 'eval "$(/opt/homebrew/bin/brew shellenv)'"' \
>> ~/.zprofile
eval "$(/opt/homebrew/bin/brew shellenv)"
```

설치 경로

<code>/opt/homebrew</code>	Apple Silicon (ARM) 접두사
<code>/usr/local</code>	Intel Mac 접두사
<code>/home/linuxbrew/.linuxbrew</code>	Linux 접두사

패키지 관리

설치 & 제거

<code>brew install git</code>	# install formula
<code>brew install git@2.40</code>	# install specific version
<code>brew uninstall git</code>	# remove formula
<code>brew reinstall git</code>	# reinstall formula

업데이트 & 업그레이드

<code>brew update</code>	# update Homebrew itself
<code>brew upgrade</code>	# upgrade all packages
<code>brew upgrade git</code>	# upgrade specific package
<code>brew pin git</code>	# prevent auto-upgrade
<code>brew unpin git</code>	# allow upgrade again

설치된 목록

<code>brew list</code>	# all installed formulae
<code>brew list --cask</code>	# all installed casks
<code>brew leaves</code>	# top-level (not dependencies)
<code>brew deps git</code>	# show dependencies of git

검색 & 정보

패키지 찾기

<code>brew search postgres</code>	# search by name
<code>brew search --cask chrome</code>	# search casks only
<code>brew info git</code>	# package details
<code>brew home git</code>	# open homepage in browser

정보 필드

Name / Version	패키지 이름 및 설치 버전
Dependencies	필수 패키지
Conflicts	공존할 수 없는 패키지
Caveats	설치 후 안내 (PATH, 설정)
Analytics	설치 횟수 (30/90/365일)

서비스

서비스 관리

<code>brew services list</code>	# all services + status
<code>brew services start postgresql</code>	# start and auto-launch
<code>brew services stop postgresql</code>	# stop service
<code>brew services restart nginx</code>	# restart service
<code>brew services run redis</code>	# start without auto-launch

서비스 메모

start	지금 실행 + 로그인 시 자동 시작 등록
run	지금만 실행 (자동 시작 없음)
stop	중지 및 자동 시작 해제
restart	중지 후 시작
list	모든 서비스와 상태 표시
Logs	~/Library/Logs/Homebrew/ 확인

Cask

GUI 앱 관리

<code>brew install --cask firefox</code>	# install GUI app
<code>brew uninstall --cask firefox</code>	# remove GUI app
<code>brew upgrade --cask</code>	# upgrade all casks
<code>brew list --cask</code>	# list installed casks

주요 Cask

google-chrome	Chrome 브라우저
visual-studio-code	VS Code 에디터
docker	Docker Desktop
iterm2	iTerm2 터미널
slack	Slack 메신저
rectangle	창 관리
1password	비밀번호 관리자
raycast	런처 (Spotlight 대체)

Tap

서드파티 저장소

<code>brew tap</code>	# list tapped repos
<code>brew tap hashicorp/tap</code>	# add a tap
<code>brew untap hashicorp/tap</code>	# remove a tap
<code>brew install hashicorp/tap/terraform</code>	# install from tap

주요 Tap

homebrew/cask	GUI 앱 (기본 포함)
homebrew/cask-fonts	폰트: <code>brew install --cask font-fira-code</code>
homebrew/bundle	Brewfile 지원 (Homebrew용 Bundler)
hashicorp/tap	Terraform, Vault, Consul

Brewfile (Bundle)

<code>brew bundle dump</code>	# generate Brewfile
<code>brew bundle install</code>	# install from Brewfile
<code>brew bundle cleanup</code>	# remove unlisted packages

정리

디스크 공간 확보

<code>brew cleanup</code>	# remove old versions
<code>brew cleanup -s</code>	# also remove cache
<code>brew cleanup --prune=all</code>	# remove all cached downloads
<code>brew autoremove</code>	# remove unused dependencies

진단

<code>brew doctor</code>	# check for issues
<code>brew config</code>	# show Homebrew config
<code>brew missing</code>	# list missing dependencies
<code>du -sh \$(brew --cache)</code>	# check cache size

일반 패턴

일상 워크플로우

Morning update	<code>brew update && brew upgrade && brew cleanup</code>
New Mac setup	Brewfile로 <code>brew bundle install</code>
Export setup	<code>brew bundle dump --file=~/.Brewfile</code>
Check health	<code>brew doctor</code>
Find what uses space	<code>brew list --formula xargs brew info</code>
Uninstall + deps	<code>brew uninstall pkg && brew autoremove</code>

Brewfile 예시

```
# ~/.Brewfile
tap "homebrew/bundle"
brew "git"
brew "node"
brew "python"
cask "visual-studio-code"
cask "docker"
```