

HOME BREW QUICK REFERENCE

Package management, casks, taps, services, cleanup

Installation

Install Homebrew

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

Post-Install (Apple Silicon)

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

Install Paths

```
/opt/homebrew Apple Silicon (ARM) prefix
/usr/local Intel Mac prefix
/home/linuxbrew/.linuxbrew Linux prefix
```

Package Management

Install & Remove

```
brew install git # install formula
brew install git@2.40 # install specific version
brew uninstall git # remove formula
brew reinstall git # reinstall formula
```

Update & Upgrade

```
brew update # update Homebrew itself
brew upgrade # upgrade all packages
brew upgrade git # upgrade specific package
brew pin git # prevent auto-upgrade
brew unpin git # allow upgrade again
```

List Installed

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

Search & Info

Finding Packages

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

Info Fields

```
Name / Version Package name and installed version
Dependencies Required packages
Conflicts Packages that cannot coexist
Caveats Post-install notes (PATH, config)
Analytics Install count (past 30/90/365 days)
```

Services

Managing Services

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

Service Notes

```
start Launch now + register for login startup
run Launch now only (no auto-start)
stop Stop and deregister from startup
restart Stop then start
list Show all services and their status
logs Check ~/Library/Logs/Homebrew/
```

Casks

GUI Application Management

```
brew install --cask firefox # install GUI app
brew uninstall --cask firefox # remove GUI app
brew upgrade --cask # upgrade all casks
brew list --cask # list installed casks
```

Common Casks

```
google-chrome Chrome browser
visual-studio-code VS Code editor
docker Docker Desktop
iterm2 iTerm2 terminal
slack Slack messaging
rectangle Window management
1password Password manager
raycast Launcher (Spotlight replacement)
```

Taps

Third-Party Repositories

```
brew tap # list tapped repos
brew tap hashicorp/tap # add a tap
brew untap hashicorp/tap # remove a tap
brew install hashicorp/tap/terraform # install from tap
```

Popular Taps

```
homebrew/cask GUI apps (included by default)
homebrew/cask-fonts Fonts: brew install --cask font-fira-code`
homebrew/bundle Brewfile support (Bundler for Homebrew)
hashicorp/tap Terraform, Vault, Consul
```

Brewfile (Bundle)

```
brew bundle dump # generate Brewfile
brew bundle install # install from Brewfile
brew bundle cleanup # remove unlisted packages
```

Cleanup

Freeing Disk Space

```
brew cleanup # remove old versions
brew cleanup -s # also remove cache
brew cleanup --prune=all # remove all cached downloads
brew autoremove # remove unused dependencies
```

Diagnostics

```
brew doctor # check for issues
brew config # show Homebrew config
brew missing # list missing dependencies
du -sh $(brew --cache) # check cache size
```

Common Patterns

Everyday Workflows

```
Morning update `brew update && brew upgrade && brew`
cleanup
```

```
New Mac setup `brew bundle install` from a Brewfile
```

```
Export setup `brew bundle dump --file=~/.Brewfile`
```

```
Check health `brew doctor`
```

```
Find what uses space `brew list --formula | xargs brew info`
```

```
Uninstall + deps `brew uninstall pkg && brew`
autoremove`
```

Brewfile Example

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