Homebrew Package manager

START HERE

`brew` **`update`**
Refresh Homebrew itself

`brew` **`list`**
See what is installed

`brew` **`search`** *`<text>`*
Find a formula or cask

`brew` **`upgrade`**
Upgrade everything outdated

`brew` **`info`** *`<name>`*
Read details first

`brew` **`uninstall`** *`<name>`*
Remove a package

`brew` **`install`** *`<name>`*
Install a package

`brew` **`cleanup`**
Free old versions and cache

Taps homebrew-core, cask → `brew update` → **Formula or cask** recipe → `brew install` → **Cellar** versioned files → `link` → **Prefix bin** on your PATH

CONCEPTS

A formula is a command-line package built or poured from a bottle; a cask is a prebuilt GUI app or binary (macOS mostly).

install and upgrade usually run update first, so a fresh brew update is rarely needed before them.

Prefix differs: /opt/homebrew on Apple Silicon, /usr/local on Intel Macs, /home/linuxbrew/.linuxbrew on Linux. Never use sudo for brew.

INSTALL AND REMOVE

`brew` **`install`** *`<formula>`*
Install a command-line package

`brew` **`install --cask`** *`<cask>`*
Install a GUI app

`brew` **`uninstall`** *`<name>`*
Remove a package

`brew` **`reinstall`** *`<name>`*
Remove and install again

`brew` **`install -n`** *`<name>`*
Dry run, change nothing

`brew` **`install --build-from-source`** *`<formula>`*
Compile instead of using a bottle
Dependencies still come as bottles when available.

`brew` **`install --HEAD`** *`<formula>`*
Build the upstream development version

`brew` **`uninstall --zap --cask`** *`<cask>`*
Also remove app support files

`brew` **`uninstall --ignore-dependencies`** *`<name>`*
Remove even if others need it

`brew` **`install --only-dependencies`** *`<formula>`*
Install just the dependencies

UPDATE AND UPGRADE

`brew` **`update`**
Fetch latest formulae and casks

`brew` **`outdated`**
List packages with newer versions

`brew` **`upgrade`**
Upgrade all outdated packages

UPDATE AND UPGRADE (cont.)

`brew` **`upgrade`** *`<name>`*
Upgrade one package

`brew` **`upgrade -n`**
Preview what would upgrade

`brew` **`upgrade --cask --greedy`**
Include casks that self-update
Casks marked auto-updates or version latest are skipped without --greedy.

`brew` **`outdated --json=v2`**
Machine-readable outdated list

`brew` **`pin`** *`<formula>`*
Hold at the current version

`brew` **`unpin`** *`<formula>`*
Allow upgrades again

`brew` **`list --pinned`**
Show pinned formulae

SEARCH AND INSPECT

`brew` **`search`** *`<text>`*
Search formulae and casks

`brew` **`search --desc`** *`<text>`*
Search descriptions too

`brew` **`info`** *`<name>`*
Version, deps, caveats, install state

`brew` **`list`**
List installed formulae and casks

`brew` **`list --versions`**
Show installed versions

`brew` **`list --cask`**
List installed casks only

`brew` **`leaves`**
Packages you installed directly

`brew` **`desc`** *`<name>`*
One-line description

`brew` **`deps --tree`** *`<name>`*
Dependency tree

`brew` **`uses --installed`** *`<name>`*
Installed packages needing this

`brew` **`home`** *`<name>`*
Open the project homepage

`brew` **`info --json=v2 --installed`**
Installed state as JSON

`brew` **`log`** *`<name>`*
Recent changes to a formula

SERVICES

`brew` **`services list`**
Services and their status

SERVICES (cont.)

`brew` **`services start`** *`<formula>`*
Run now and at login/boot

`brew` **`services stop`** *`<formula>`*
Stop and unregister

`brew` **`services restart`** *`<formula>`*
Stop and start again

`brew` **`services run`** *`<formula>`*
Run now, not at login/boot

`sudo brew services start` *`<formula>`*
System-wide service as root

`brew` **`services cleanup`**
Remove unused service files

Without sudo a service is per-user (launchd or systemd user unit); with sudo it is system-wide.

TAPS

`brew` **`tap`**
List added taps

`brew` **`tap`** *`<user>`*`/`*`<repo>`*
Add a third-party repository

`brew` **`untap`** *`<user>`*`/`*`<repo>`*
Remove a tap

`brew` **`tap-info`** *`<user>`*`/`*`<repo>`*
Show tap details

`brew` **`install`** *`<user>`*`/`*`<repo>`*`/`*`<formula>`*
Install from a tap in one step

CLEANUP AND DOCTOR

`brew` **`cleanup`**
Delete old versions and stale downloads

`brew` **`cleanup -n`**
Show what cleanup would delete

`brew` **`autoremove`**
Remove unneeded dependencies

`brew` **`doctor`**
Diagnose common problems

`brew` **`config`**
Print setup info for bug reports

`brew` **`cleanup --prune=`***`<days>`*
Also drop cache older than N days

`brew` **`--prefix`**
Show the install prefix

`brew` **`--cache`**
Show the download cache path

CLEANUP AND DOCTOR (cont.)

`brew` **`missing`**
Check for missing dependencies

`brew` **`link`** *`<formula>`*
Link a formula into the prefix

`brew` **`unlink`** *`<formula>`*
Unlink but keep installed

Install and upgrade run a cleanup afterwards; export HOMEBREW_NO_INSTALL_CLEANUP=1 to skip it.

BREWFILE (BUNDLE)

`brew` **`bundle dump`**
Write installed state to Brewfile

`brew` **`bundle`**
Install everything in the Brewfile

`brew` **`bundle check`**
Are all entries satisfied?

`brew` **`bundle list`**
List what the Brewfile declares

`brew` **`bundle cleanup`**
Show extras not in the Brewfile
Only previews; add --force to actually uninstall them.

`brew` **`bundle cleanup --force`**
Uninstall everything not listed

`brew` **`bundle --file=`***`<path>`*
Use a specific Brewfile

`brew` **`bundle dump --force`**
Overwrite an existing Brewfile

Brewfile

```
tap "<user>/<repo>"
brew "<formula>"
cask "<cask>"
```

RECIPES

Weekly maintenance

```
brew update
brew outdated
brew upgrade
brew autoremove
brew cleanup
```

Capture and restore a setup

```
brew bundle dump --file=~/
Brewfile
# on the new machine
brew bundle --file=~/Brewfile
```

Why is this installed?

```
brew leaves | grep <name>
brew uses --installed <name>
brew deps --tree <name>
```

Hold one package back

```
brew pin <formula>
brew upgrade   # skips it
brew unpin <formula>
```