NAME

brew – The Missing Package Manager for macOS (or Linux)

SYNOPSIS

brew --version
brew *command* [**--verbose**|**-v**] [*options*] [*formula*] ...

DESCRIPTION

Homebrew is the easiest and most flexible way to install the UNIX tools Apple didn´t include with macOS. It can also install software not packaged for your Linux distribution to your home directory without requiring **sudo**.

TERMINOLOGY

formula
Homebrew package definition built from upstream sources

cask
Homebrew package definition that installs macOS native applications

keg
installation destination directory of a given **formula** version e.g. **/usr/local/Cellar/foo/0.1**

rack
directory containing one or more versioned kegs e.g. **/usr/local/Cellar/foo**

keg-only
a **formula** is **keg-only** if it is not symlinked into Homebrew´s prefix (e.g. **/usr/local**)

cellar
directory containing one or more named **racks** e.g. **/usr/local/Cellar**

Caskroom
directory containing one or more named **casks** e.g. **/usr/local/Caskroom**

external command
brew subcommand defined outside of the Homebrew/brew GitHub repository

tap
directory (and usually Git repository) of **formulae**, **casks** and/or **external commands**

bottle
pre-built **keg** poured into the **cellar**/**rack** instead of building from upstream sources

ESSENTIAL COMMANDS

For the full command list, see the *COMMANDS* section.

With **--verbose** or **--debug**, many commands print extra debugging information. Note that these options should only appear after a command.

Some command behaviour can be customised with environment variables; see the *ENVIRONMENT* section.

install *formula*
Install *formula*.

formula is usually the name of the formula to install, but it has other syntaxes which are listed in the *SPECIFYING FORMULAE* section.

uninstall *formula*
Uninstall *formula*.

list
List all installed formulae.

search [*text*|/*text*/]
Perform a substring search of cask tokens and formula names for *text*. If *text* is flanked by slashes, it is interpreted as a regular expression. The search for *text* is extended online to **homebrew/core** and

homebrew/cask. If no search term is provided, all locally available formulae are listed.

COMMANDS

analytics [*subcommand*]

Control Homebrew´s anonymous aggregate user behaviour analytics. Read more at *https://docs.brew.sh/Analytics*.

brew analytics [**state**]
: Display the current state of Homebrew´s analytics.

brew analytics (**on**|**off**)
: Turn Homebrew´s analytics on or off respectively.

brew analytics regenerate−uuid
: Regenerate the UUID used for Homebrew´s analytics.

autoremove [*−−dry−run*]

Uninstall formulae that were only installed as a dependency of another formula and are now no longer needed.

−n, **−−dry−run**
: List what would be uninstalled, but do not actually uninstall anything.

casks

List all locally installable casks including short names.

cleanup [*options*] [*formula*|*cask* ...]

Remove stale lock files and outdated downloads for all formulae and casks, and remove old versions of installed formulae. If arguments are specified, only do this for the given formulae and casks. Removes all downloads more than 120 days old. This can be adjusted with **HOMEBREW_CLEANUP_MAX_AGE_DAYS**.

−−prune
: Remove all cache files older than specified *days*. If you want to remove everything, use **−−prune=all**.

−n, **−−dry−run**
: Show what would be removed, but do not actually remove anything.

−s
: Scrub the cache, including downloads for even the latest versions. Note that downloads for any installed formulae or casks will still not be deleted. If you want to delete those too: **rm −rf "$(brew −−cache)"**

−−prune−prefix
: Only prune the symlinks and directories from the prefix and remove no other files.

commands [*−−quiet*] [*−−include−aliases*]

Show lists of built−in and external commands.

−q, **−−quiet**
: List only the names of commands without category headers.

−−include−aliases
: Include aliases of internal commands.

completions [*subcommand*]

Control whether Homebrew automatically links external tap shell completion files. Read more at *https://docs.brew.sh/Shell−Completion*.

brew completions [**state**]
: Display the current state of Homebrew´s completions.

brew completions (**link**|**unlink**)
: Link or unlink Homebrew´s completions.

config, **−−config**

Show Homebrew and system configuration info useful for debugging. If you file a bug report, you will be required to provide this information.

deps [*options*] [*formula*|*cask* ...]

Show dependencies for *formula*. Additional options specific to *formula* may be appended to the command. When given multiple formula arguments, show the intersection of dependencies for each formula.

−n Sort dependencies in topological order.

−−1 Only show dependencies one level down, instead of recursing.

−−union
Show the union of dependencies for multiple *formula*, instead of the intersection.

−−full−name
List dependencies by their full name.

−−include−build
Include **:build** dependencies for *formula*.

−−include−optional
Include **:optional** dependencies for *formula*.

−−include−test
Include **:test** dependencies for *formula* (non−recursive).

−−skip−recommended
Skip **:recommended** dependencies for *formula*.

−−include−requirements
Include requirements in addition to dependencies for *formula*.

−−tree Show dependencies as a tree. When given multiple formula arguments, show individual trees for each formula.

−−graph
Show dependencies as a directed graph.

−−dot Show text−based graph description in DOT format.

−−annotate
Mark any build, test, optional, or recommended dependencies as such in the output.

−−installed
List dependencies for formulae that are currently installed. If *formula* is specified, list only its dependencies that are currently installed.

−−all List dependencies for all available formulae.

−−for−each
Switch into the mode used by the **−−all** option, but only list dependencies for each provided *formula*, one formula per line. This is used for debugging the **−−installed**/**−−all** display mode.

−−formula
Treat all named arguments as formulae.

−−cask Treat all named arguments as casks.

desc [*options*] *formula*|*text*|/*regex*/ [...]

Display *formula*´s name and one−line description. Formula descriptions are cached; the cache is created on the first search, making that search slower than subsequent ones.

−s, **−−search**
Search both names and descriptions for *text*. If *text* is flanked by slashes, it is interpreted as a regular expression.

−n, **−−name**
Search just names for *text*. If *text* is flanked by slashes, it is interpreted as a regular expression.

−d, **−−description**
Search just descriptions for *text*. If *text* is flanked by slashes, it is interpreted as a regular expression.

developer [*subcommand*]
Control Homebrew´s developer mode. When developer mode is enabled, **brew update** will update Homebrew to the latest commit on the **master** branch instead of the latest stable version along with some other behaviour changes.

brew developer [**state**]
Display the current state of Homebrew´s developer mode.

brew developer (**on**|**off**)
Turn Homebrew´s developer mode on or off respectively.

doctor, **dr** [*−−list−checks*] [*−−audit−debug*] [*diagnostic_check* ...]
Check your system for potential problems. Will exit with a non−zero status if any potential problems are found. Please note that these warnings are just used to help the Homebrew maintainers with debugging if you file an issue. If everything you use Homebrew for is working fine: please don´t worry or file an issue; just ignore this.

−−list−checks
List all audit methods, which can be run individually if provided as arguments.

−D, **−−audit−debug**
Enable debugging and profiling of audit methods.

fetch [*options*] *formula*|*cask* [...]
Download a bottle (if available) or source packages for *formula*e and binaries for *cask*s. For files, also print SHA−256 checksums.

−−bottle−tag
Download a bottle for given tag.

−−HEAD
Fetch HEAD version instead of stable version.

−f, **−−force**
Remove a previously cached version and re−fetch.

−v, **−−verbose**
Do a verbose VCS checkout, if the URL represents a VCS. This is useful for seeing if an existing VCS cache has been updated.

−−retry
Retry if downloading fails or re−download if the checksum of a previously cached version no longer matches.

−−deps Also download dependencies for any listed *formula*.

−s, **−−build−from−source**
Download source packages rather than a bottle.

−−build−bottle
Download source packages (for eventual bottling) rather than a bottle.

−−force−bottle
Download a bottle if it exists for the current or newest version of macOS, even if it would not be used during installation.

−−[no−]quarantine
Disable/enable quarantining of downloads (default: enabled).

−−formula
Treat all named arguments as formulae.

−−cask Treat all named arguments as casks.

formulae
List all locally installable formulae including short names.

gist−logs [*options*] *formula*
Upload logs for a failed build of *formula* to a new Gist. Presents an error message if no logs are found.

−−with−hostname
Include the hostname in the Gist.

−n, **−−new−issue**
Automatically create a new issue in the appropriate GitHub repository after creating the Gist.

−p, **−−private**
The Gist will be marked private and will not appear in listings but will be accessible with its link.

home, **homepage** [*−−formula*] [*−−cask*] [*formula*|*cask* ...]
Open a *formula* or *cask*´s homepage in a browser, or open Homebrew´s own homepage if no argument is provided.

−−formula
Treat all named arguments as formulae.

−−cask Treat all named arguments as casks.

info, **abv** [*options*] [*formula*|*cask* ...]
Display brief statistics for your Homebrew installation.

If a *formula* or *cask* is provided, show summary of information about it.

−−analytics
List global Homebrew analytics data or, if specified, installation and build error data for *formula* (provided neither **HOMEBREW_NO_ANALYTICS** nor **HOMEBREW_NO_GITHUB_API** are set).

−−days How many days of analytics data to retrieve. The value for *days* must be **30**, **90** or **365**. The default is **30**.

−−category
Which type of analytics data to retrieve. The value for *category* must be **install**, **install−on−request** or **build−error**; **cask−install** or **os−version** may be specified if *formula* is not. The default is **install**.

−−github
Open the GitHub source page for *formula* and *cask* in a browser. To view the history locally: **brew log −p** *formula* or *cask*

−−json Print a JSON representation. Currently the default value for *version* is **v1** for *formula*. For *formula* and *cask* use **v2**. See the docs for examples of using the JSON output: *https://docs.brew.sh/Querying−Brew*

−−installed
Print JSON of formulae that are currently installed.

−−all Print JSON of all available formulae.

−v, **−−verbose**
Show more verbose analytics data for *formula*.

−−formula
Treat all named arguments as formulae.

−−cask Treat all named arguments as casks.

install [*options*] *formula*|*cask* [...]
Install a *formula* or *cask*. Additional options specific to a *formula* may be appended to the command.

Unless **HOMEBREW_NO_INSTALLED_DEPENDENTS_CHECK** is set, **brew upgrade** or **brew reinstall** will be run for outdated dependents and dependents with broken linkage, respectively.

Unless **HOMEBREW_NO_INSTALL_CLEANUP** is set, **brew cleanup** will then be run for the installed formulae or, every 30 days, for all formulae.

Unless **HOMEBREW_NO_INSTALL_UPGRADE** is set, **brew install <formula>** will upgrade *formula* if it is already installed but outdated.

−d, **−−debug**
If brewing fails, open an interactive debugging session with access to IRB or a shell inside the temporary build directory.

−f, **−−force**
Install formulae without checking for previously installed keg−only or non−migrated versions. When installing casks, overwrite existing files (binaries and symlinks are excluded, unless originally from the same cask).

−v, **−−verbose**
Print the verification and postinstall steps.

−−formula
Treat all named arguments as formulae.

−−ignore−dependencies
An unsupported Homebrew development flag to skip installing any dependencies of any kind. If the dependencies are not already present, the formula will have issues. If you´re not developing Homebrew, consider adjusting your PATH rather than using this flag.

−−only−dependencies
Install the dependencies with specified options but do not install the formula itself.

−−cc Attempt to compile using the specified *compiler*, which should be the name of the compiler´s executable, e.g. **gcc−7** for GCC 7. In order to use LLVM´s clang, specify **llvm_clang**. To use the Apple−provided clang, specify **clang**. This option will only accept compilers that are provided by Homebrew or bundled with macOS. Please do not file issues if you encounter errors while using this option.

−s, **−−build−from−source**
Compile *formula* from source even if a bottle is provided. Dependencies will still be installed from bottles if they are available.

−−force−bottle
Install from a bottle if it exists for the current or newest version of macOS, even if it would not normally be used for installation.

−−include−test
Install testing dependencies required to run **brew test** *formula*.

−−HEAD
If *formula* defines it, install the HEAD version, aka. main, trunk, unstable, master.

−−fetch−HEAD
Fetch the upstream repository to detect if the HEAD installation of the formula is outdated. Otherwise, the repository´s HEAD will only be checked for updates when a new stable or development version has been released.

−−keep−tmp
Retain the temporary files created during installation.

−−build−bottle
Prepare the formula for eventual bottling during installation, skipping any post−install steps.

−−bottle−arch
Optimise bottles for the specified architecture rather than the oldest architecture supported by the version of macOS the bottles are built on.

−−display−times
Print install times for each package at the end of the run.

−i, **−−interactive**
Download and patch *formula*, then open a shell. This allows the user to run **./configure −−help** and otherwise determine how to turn the software package into a Homebrew package.

−g, **−−git**
Create a Git repository, useful for creating patches to the software.

−−overwrite
Delete files that already exist in the prefix while linking.

−−cask Treat all named arguments as casks.

−−[no−]binaries
Disable/enable linking of helper executables (default: enabled).

−−require−sha
Require all casks to have a checksum.

−−[no−]quarantine
Disable/enable quarantining of downloads (default: enabled).

−−skip−cask−deps
Skip installing cask dependencies.

leaves [*−−installed−on−request*] [*−−installed−as−dependency*]
List installed formulae that are not dependencies of another installed formula.

−r, **−−installed−on−request**
Only list leaves that were manually installed.

−p, **−−installed−as−dependency**
Only list leaves that were installed as dependencies.

link, **ln** [*options*] *installed_formula* [...]
Symlink all of *formula*´s installed files into Homebrew´s prefix. This is done automatically when you install formulae but can be useful for DIY installations.

−−overwrite
Delete files that already exist in the prefix while linking.

−n, **−−dry−run**
List files which would be linked or deleted by **brew link −−overwrite** without actually linking or deleting any files.

−f, **−−force**
Allow keg−only formulae to be linked.

−−HEAD
Link the HEAD version of the formula if it is installed.

list, **ls** [*options*] [*installed_formula*|*installed_cask* ...]
List all installed formulae and casks.

If *formula* is provided, summarise the paths within its current keg. If *cask* is provided, list its artifacts.

−−formula
List only formulae, or treat all named arguments as formulae.

−−cask List only casks, or treat all named arguments as casks.

−−full−name
Print formulae with fully−qualified names. Unless **−−full−name**, **−−versions** or **−−pinned** are passed, other options (i.e. **−1**, **−l**, **−r** and **−t**) are passed to **ls**(1) which produces the actual output.

−−versions
Show the version number for installed formulae, or only the specified formulae if *formula* are provided.

−−multiple
Only show formulae with multiple versions installed.

−−pinned
List only pinned formulae, or only the specified (pinned) formulae if *formula* are provided. See also **pin**, **unpin**.

−1 Force output to be one entry per line. This is the default when output is not to a terminal.

−l List formulae and/or casks in long format. Has no effect when a formula or cask name is passed as an argument.

−r Reverse the order of the formulae and/or casks sort to list the oldest entries first. Has no effect when a formula or cask name is passed as an argument.

−t Sort formulae and/or casks by time modified, listing most recently modified first. Has no effect when a formula or cask name is passed as an argument.

log [*options*] [*formula*|*cask*]
Show the **git log** for *formula* or *cask*, or show the log for the Homebrew repository if no formula or cask is provided.

−p, **−−patch**
Also print patch from commit.

−−stat Also print diffstat from commit.

−−oneline
Print only one line per commit.

−1 Print only one commit.

−n, **−−max−count**
Print only a specified number of commits.

−−formula
Treat all named arguments as formulae.

−−cask Treat all named arguments as casks.

migrate [*−−force*] [*−−dry−run*] *installed_formula* [...]
Migrate renamed packages to new names, where *formula* are old names of packages.

−f, **−−force**
Treat installed *formula* and provided *formula* as if they are from the same taps and migrate them anyway.

−n, **−−dry−run**
Show what would be migrated, but do not actually migrate anything.

missing [*−−hide*=] [*formula* ...]
Check the given *formula* kegs for missing dependencies. If no *formula* are provided, check all kegs. Will exit with a non−zero status if any kegs are found to be missing dependencies.

−−hide Act as if none of the specified *hidden* are installed. *hidden* should be a comma−separated list of formulae.

options [*options*] [*formula* ...]
Show install options specific to *formula*.

−−compact
Show all options on a single line separated by spaces.

−−installed
Show options for formulae that are currently installed.

−−all Show options for all available formulae.

−−command
Show options for the specified *command*.

outdated [*options*] [*formula*|*cask* ...]
List installed casks and formulae that have an updated version available. By default, version information is displayed in interactive shells, and suppressed otherwise.

−q, **−−quiet**
List only the names of outdated kegs (takes precedence over **−−verbose**).

−v, **−−verbose**
Include detailed version information.

−−formula
List only outdated formulae.

−−cask List only outdated casks.

−−json Print output in JSON format. There are two versions: **v1** and **v2**. **v1** is deprecated and is currently the default if no version is specified. **v2** prints outdated formulae and casks.

−−fetch−HEAD
Fetch the upstream repository to detect if the HEAD installation of the formula is outdated. Otherwise, the repository´s HEAD will only be checked for updates when a new stable or development version has been released.

−−greedy
Print outdated casks with **auto_updates true** or **version :latest**.

−−greedy−latest
Print outdated casks including those with **version :latest**.

−−greedy−auto−updates
Print outdated casks including those with **auto_updates true**.

pin *installed_formula* [...]
Pin the specified *formula*, preventing them from being upgraded when issuing the **brew upgrade** *formula* command. See also **unpin**.

postinstall *installed_formula* [...]
Rerun the post−install steps for *formula*.

readall [*−−aliases*] [*−−syntax*] [*tap* ...]
Import all items from the specified *tap*, or from all installed taps if none is provided. This can be useful for debugging issues across all items when making significant changes to **formula.rb**, testing the performance of loading all items or checking if any current formulae/casks have Ruby issues.

−−aliases
Verify any alias symlinks in each tap.

−−syntax
Syntax−check all of Homebrew´s Ruby files (if no **<tap>** is passed).

reinstall [*options*] *formula*|*cask* [...]
Uninstall and then reinstall a *formula* or *cask* using the same options it was originally installed with, plus any appended options specific to a *formula*.

Unless **HOMEBREW_NO_INSTALLED_DEPENDENTS_CHECK** is set, **brew upgrade** or **brew reinstall** will be run for outdated dependents and dependents with broken linkage, respectively.

Unless **HOMEBREW_NO_INSTALL_CLEANUP** is set, **brew cleanup** will then be run for the reinstalled formulae or, every 30 days, for all formulae.

−d, **−−debug**
If brewing fails, open an interactive debugging session with access to IRB or a shell inside the temporary build directory.

−f, **−−force**
Install without checking for previously installed keg−only or non−migrated versions.

−v, **−−verbose**
Print the verification and postinstall steps.

−−formula
Treat all named arguments as formulae.

−s, **−−build−from−source**
Compile *formula* from source even if a bottle is available.

−i, **−−interactive**
Download and patch *formula*, then open a shell. This allows the user to run **./configure −−help** and otherwise determine how to turn the software package into a Homebrew package.

−−force−bottle
Install from a bottle if it exists for the current or newest version of macOS, even if it would not normally be used for installation.

−−keep−tmp
Retain the temporary files created during installation.

−−display−times
Print install times for each formula at the end of the run.

−g, **−−git**
Create a Git repository, useful for creating patches to the software.

−−cask Treat all named arguments as casks.

−−[no−]binaries
Disable/enable linking of helper executables (default: enabled).

−−require−sha
Require all casks to have a checksum.

−−[no−]quarantine
Disable/enable quarantining of downloads (default: enabled).

−−skip−cask−deps
Skip installing cask dependencies.

search, **−S** [*options*] *text*|/*regex*/ [...]
Perform a substring search of cask tokens and formula names for *text*. If *text* is flanked by slashes, it is interpreted as a regular expression. The search for *text* is extended online to **homebrew/core** and **homebrew/cask**.

−−formula
Search online and locally for formulae.

−−cask Search online and locally for casks.

−−desc Search for formulae with a description matching *text* and casks with a name matching *text*.

−−pull−request
Search for GitHub pull requests containing *text*.

−−open
Search for only open GitHub pull requests.

−−closed
Search for only closed GitHub pull requests.

−−repology
Search for *text* in the given database.

−−macports
Search for *text* in the given database.

−−fink Search for *text* in the given database.

−−opensuse
Search for *text* in the given database.

−−fedora
Search for *text* in the given database.

−−archlinux
Search for *text* in the given database.

−−debian
Search for *text* in the given database.

−−ubuntu
Search for *text* in the given database.

shellenv
Print export statements. When run in a shell, this installation of Homebrew will be added to your **PATH**, **MANPATH**, and **INFOPATH**.

The variables **HOMEBREW_PREFIX**, **HOMEBREW_CELLAR** and **HOMEBREW_REPOSITORY** are also exported to avoid querying them multiple times. To help guarantee idempotence, this command produces no output when Homebrew´s **bin** and **sbin** directories are first and second respectively in your **PATH**. Consider adding evaluation of this command´s output to your dotfiles (e.g. **˜/.profile**, **˜/.bash_profile**, or **˜/.zprofile**) with: **eval "$(brew shellenv)"**

tap [*options*] [*user*/*repo*] [*URL*]
Tap a formula repository.

If no arguments are provided, list all installed taps.

With *URL* unspecified, tap a formula repository from GitHub using HTTPS. Since so many taps are hosted on GitHub, this command is a shortcut for **brew tap** *user*/*repo* **https://github.com/***user***/homebrew−***repo*.

With *URL* specified, tap a formula repository from anywhere, using any transport protocol that **git**(1) handles. The one−argument form of **tap** simplifies but also limits. This two−argument command makes no assumptions, so taps can be cloned from places other than GitHub and using protocols other than HTTPS, e.g. SSH, git, HTTP, FTP(S), rsync.

−−force−auto−update
Auto−update tap even if it is not hosted on GitHub. By default, only taps hosted on GitHub are auto−updated (for performance reasons).

−−custom−remote
Install or change a tap with a custom remote. Useful for mirrors.

−−repair
Migrate tapped formulae from symlink−based to directory−based structure.

−−list−pinned
List all pinned taps.

tap−info [*−−installed*] [*−−json*] [*tap* ...]
Show detailed information about one or more *tap*s.

If no *tap* names are provided, display brief statistics for all installed taps.

−−installed
Show information on each installed tap.

−−json Print a JSON representation of *tap*. Currently the default and only accepted value for *version* is **v1**. See the docs for examples of using the JSON output: *https://docs.brew.sh/Querying−Brew*

uninstall, **remove**, **rm** [*options*] *installed_formula*|*installed_cask* [...]
Uninstall a *formula* or *cask*.

−f, **−−force**
Delete all installed versions of *formula*. Uninstall even if *cask* is not installed, overwrite existing files and ignore errors when removing files.

−−zap Remove all files associated with a *cask*. *May remove files which are shared between applications*.

−−ignore−dependencies
Don´t fail uninstall, even if *formula* is a dependency of any installed formulae.

−−formula
Treat all named arguments as formulae.

−−cask Treat all named arguments as casks.

unlink [*−−dry−run*] *installed_formula* [...]
Remove symlinks for *formula* from Homebrew´s prefix. This can be useful for temporarily disabling a formula: **brew unlink** *formula* **&&** *commands* **&& brew link** *formula*

−n, **−−dry−run**
List files which would be unlinked without actually unlinking or deleting any files.

unpin *installed_formula* [...]
Unpin *formula*, allowing them to be upgraded by **brew upgrade** *formula*. See also **pin**.

untap [*−−force*] *tap* [...]
Remove a tapped formula repository.

−f, **−−force**
Untap even if formulae or casks from this tap are currently installed.

update [*options*]
Fetch the newest version of Homebrew and all formulae from GitHub using **git**(1) and perform any necessary migrations.

−−merge
Use **git merge** to apply updates (rather than **git rebase**).

−−preinstall
Run on auto−updates (e.g. before **brew install**). Skips some slower steps.

−f, **−−force**
Always do a slower, full update check (even if unnecessary).

update−reset [*repository* ...]

Fetch and reset Homebrew and all tap repositories (or any specified *repository*) using **git**(1) to their latest **origin/HEAD**.

Note: this will destroy all your uncommitted or committed changes.

upgrade [*options*] [*outdated_formula*|*outdated_cask* ...]

Upgrade outdated casks and outdated, unpinned formulae using the same options they were originally installed with, plus any appended brew formula options. If *cask* or *formula* are specified, upgrade only the given *cask* or *formula* kegs (unless they are pinned; see **pin**, **unpin**).

Unless **HOMEBREW_NO_INSTALLED_DEPENDENTS_CHECK** is set, **brew upgrade** or **brew reinstall** will be run for outdated dependents and dependents with broken linkage, respectively.

Unless **HOMEBREW_NO_INSTALL_CLEANUP** is set, **brew cleanup** will then be run for the upgraded formulae or, every 30 days, for all formulae.

−d, **−−debug**
If brewing fails, open an interactive debugging session with access to IRB or a shell inside the temporary build directory.

−f, **−−force**
Install formulae without checking for previously installed keg−only or non−migrated versions. When installing casks, overwrite existing files (binaries and symlinks are excluded, unless originally from the same cask).

−v, **−−verbose**
Print the verification and postinstall steps.

−n, **−−dry−run**
Show what would be upgraded, but do not actually upgrade anything.

−−formula
Treat all named arguments as formulae. If no named arguments are specified, upgrade only outdated formulae.

−s, **−−build−from−source**
Compile *formula* from source even if a bottle is available.

−i, **−−interactive**
Download and patch *formula*, then open a shell. This allows the user to run **./configure −−help** and otherwise determine how to turn the software package into a Homebrew package.

−−force−bottle
Install from a bottle if it exists for the current or newest version of macOS, even if it would not normally be used for installation.

−−fetch−HEAD
Fetch the upstream repository to detect if the HEAD installation of the formula is outdated. Otherwise, the repository´s HEAD will only be checked for updates when a new stable or development version has been released.

−−ignore−pinned
Set a successful exit status even if pinned formulae are not upgraded.

−−keep−tmp
Retain the temporary files created during installation.

−−display−times
Print install times for each package at the end of the run.

−−cask Treat all named arguments as casks. If no named arguments are specified, upgrade only outdated casks.

−−[no−]binaries
Disable/enable linking of helper executables (default: enabled).

−−require−sha
Require all casks to have a checksum.

−−[no−]quarantine
Disable/enable quarantining of downloads (default: enabled).

−−skip−cask−deps
Skip installing cask dependencies.

−−greedy
Also include casks with **auto_updates true** or **version :latest**.

−−greedy−latest
Also include casks with **version :latest**.

−−greedy−auto−updates
Also include casks with **auto_updates true**.

uses [*options*] *formula* [...]
Show formulae and casks that specify *formula* as a dependency; that is, show dependents of *formula*. When given multiple formula arguments, show the intersection of formulae that use *formula*. By default, **uses** shows all formulae and casks that specify *formula* as a required or recommended dependency for their stable builds.

−−recursive
Resolve more than one level of dependencies.

−−installed
Only list formulae and casks that are currently installed.

−−include−build
Include all formulae that specify *formula* as **:build** type dependency.

−−include−test
Include all formulae that specify *formula* as **:test** type dependency.

−−include−optional
Include all formulae that specify *formula* as **:optional** type dependency.

−−skip−recommended
Skip all formulae that specify *formula* as **:recommended** type dependency.

−−formula
Include only formulae.

−−cask Include only casks.

−−cache [*options*] [*formula*|*cask* ...]
Display Homebrew´s download cache. See also **HOMEBREW_CACHE**.

If *formula* is provided, display the file or directory used to cache *formula*.

−s, **−−build−from−source**
Show the cache file used when building from source.

−−force−bottle
Show the cache file used when pouring a bottle.

−−bottle−tag
Show the cache file used when pouring a bottle for the given tag.

−−HEAD
Show the cache file used when building from HEAD.

−−formula
Only show cache files for formulae.

−−cask Only show cache files for casks.

−−caskroom [*cask* ...]
Display Homebrew´s Caskroom path.

If *cask* is provided, display the location in the Caskroom where *cask* would be installed, without any sort of versioned directory as the last path.

−−cellar [*formula* ...]
Display Homebrew´s Cellar path. *Default:* **$(brew −−prefix)/Cellar**, or if that directory doesn´t exist, **$(brew −−repository)/Cellar**.

If *formula* is provided, display the location in the Cellar where *formula* would be installed, without any sort of versioned directory as the last path.

−−env, **environment** [−−*shell*=] [−−*plain*] [*formula* ...]
Summarise Homebrew´s build environment as a plain list.

If the command´s output is sent through a pipe and no shell is specified, the list is formatted for export to **bash**(1) unless **−−plain** is passed.

−−shell Generate a list of environment variables for the specified shell, or **−−shell=auto** to detect the current shell.

−−plain
Generate plain output even when piped.

−−prefix [−−*unbrewed*] [−−*installed*] [*formula* ...]
Display Homebrew´s install path. *Default:*

- macOS Intel: **/usr/local**
- macOS ARM: **/opt/homebrew**
- Linux: **/home/linuxbrew/.linuxbrew**

If *formula* is provided, display the location where *formula* is or would be installed.

−−unbrewed
List files in Homebrew´s prefix not installed by Homebrew.

−−installed
Outputs nothing and returns a failing status code if *formula* is not installed.

−−repository, **−−repo** [*tap* ...]
Display where Homebrew´s git repository is located.

If *user*/*repo* are provided, display where tap *user*/*repo*´s directory is located.

−−version, **−v**
Print the version numbers of Homebrew, Homebrew/homebrew−core and Homebrew/homebrew−cask (if tapped) to standard output.

DEVELOPER COMMANDS

audit [*options*] [*formula*|*cask* ...]
Check *formula* for Homebrew coding style violations. This should be run before submitting a new formula or cask. If no *formula*|*cask* are provided, check all locally available formulae and casks and skip style checks. Will exit with a non−zero status if any errors are found.

−−strict
Run additional, stricter style checks.

−−git Run additional, slower style checks that navigate the Git repository.

−−online
Run additional, slower style checks that require a network connection.

−−installed
Only check formulae and casks that are currently installed.

−−new Run various additional style checks to determine if a new formula or cask is eligible for Homebrew. This should be used when creating new formula and implies **−−strict** and **−−online**.

−−[no−]appcast
Audit the appcast.

−−token−conflicts
Audit for token conflicts.

−−tap Check the formulae within the given tap, specified as *user*/*repo*.

−−fix Fix style violations automatically using RuboCop´s auto−correct feature.

−−display−cop−names
Include the RuboCop cop name for each violation in the output.

−−display−filename
Prefix every line of output with the file or formula name being audited, to make output easy to grep.

−−display−failures−only
Only display casks that fail the audit. This is the default for formulae.

−−skip−style
Skip running non−RuboCop style checks. Useful if you plan on running **brew style** separately. Enabled by default unless a formula is specified by name.

−D, **−−audit−debug**
Enable debugging and profiling of audit methods.

−−only Specify a comma−separated *method* list to only run the methods named **audit_***method*.

−−except
Specify a comma−separated *method* list to skip running the methods named **audit_***method*.

−−only−cops
Specify a comma−separated *cops* list to check for violations of only the listed RuboCop cops.

−−except−cops
Specify a comma−separated *cops* list to skip checking for violations of the listed RuboCop cops.

−−formula
Treat all named arguments as formulae.

−−cask Treat all named arguments as casks.

bottle [*options*] *installed_formula*|*file* [...]
Generate a bottle (binary package) from a formula that was installed with **−−build−bottle**. If the formula specifies a rebuild version, it will be incremented in the generated DSL. Passing **−−keep−old** will attempt to keep it at its original value, while **−−no−rebuild** will remove it.

−−skip−relocation
Do not check if the bottle can be marked as relocatable.

−−force−core−tap
Build a bottle even if *formula* is not in **homebrew/core** or any installed taps.

−−no−rebuild
If the formula specifies a rebuild version, remove it from the generated DSL.

−−keep−old
If the formula specifies a rebuild version, attempt to preserve its value in the generated DSL.

−−json Write bottle information to a JSON file, which can be used as the value for **−−merge**.

−−merge
Generate an updated bottle block for a formula and optionally merge it into the formula file. Instead of a formula name, requires the path to a JSON file generated with **brew bottle −−json** *formula*.

−−write
Write changes to the formula file. A new commit will be generated unless **−−no−commit** is passed.

−−no−commit
When passed with **−−write**, a new commit will not generated after writing changes to the formula file.

−−only−json−tab
When passed with **−−json**, the tab will be written to the JSON file but not the bottle.

−−committer
Specify a committer name and email in **git**´s standard author format.

−−root−url
Use the specified *URL* as the root of the bottle´s URL instead of Homebrew´s default.

−−root−url−using
Use the specified download strategy class for downloading the bottle´s URL instead of Homebrew´s default.

bump [*options*] [*formula*|*cask* ...]
Display out−of−date brew formulae and the latest version available. If the returned current and livecheck versions differ or when querying specific formulae, also displays whether a pull request has been opened with the URL.

−−full−name
Print formulae/casks with fully−qualified names.

−−no−pull−requests
Do not retrieve pull requests from GitHub.

−−formula
Check only formulae.

−−cask Check only casks.

−−open−pr
Open a pull request for the new version if there are none already open.

−−limit
Limit number of package results returned.

−−start−with
Letter or word that the list of package results should alphabetically follow.

bump−cask−pr [*options*] *cask*
Create a pull request to update *cask* with a new version.

A best effort to determine the *SHA−256* will be made if the value is not supplied by the user.

−n, **−−dry−run**
Print what would be done rather than doing it.

−−write−only
Make the expected file modifications without taking any Git actions.

−−commit
When passed with **−−write−only**, generate a new commit after writing changes to the cask file.

−−no−audit
Don´t run **brew audit** before opening the PR.

−−online
Run **brew audit −−online** before opening the PR.

−−no−style
Don´t run **brew style −−fix** before opening the PR.

−−no−browse
Print the pull request URL instead of opening in a browser.

−−no−fork
Don´t try to fork the repository.

−−version
Specify the new *version* for the cask.

−−message
Append *message* to the default pull request message.

−−url Specify the *URL* for the new download.

−−sha256
Specify the *SHA−256* checksum of the new download.

−−fork−org
Use the specified GitHub organization for forking.

−f, **−−force**
Ignore duplicate open PRs.

bump−formula−pr [*options*] [*formula*]
Create a pull request to update *formula* with a new URL or a new tag.

If a *URL* is specified, the *SHA−256* checksum of the new download should also be specified. A best effort to determine the *SHA−256* and *formula* name will be made if either or both values are not supplied by the user.

If a *tag* is specified, the Git commit *revision* corresponding to that tag should also be specified. A best effort to determine the *revision* will be made if the value is not supplied by the user.

If a *version* is specified, a best effort to determine the *URL* and *SHA−256* or the *tag* and *revision* will be made if both values are not supplied by the user.

Note: this command cannot be used to transition a formula from a URL−and−SHA−256 style specification into a tag−and−revision style specification, nor vice versa. It must use whichever style specification the formula already uses.

−n, **−−dry−run**
Print what would be done rather than doing it.

−−write−only
Make the expected file modifications without taking any Git actions.

−−commit
When passed with **−−write−only**, generate a new commit after writing changes to the formula file.

−−no−audit
Don´t run **brew audit** before opening the PR.

−−strict
Run **brew audit −−strict** before opening the PR.

−−online
Run **brew audit −−online** before opening the PR.

−−no−browse
Print the pull request URL instead of opening in a browser.

−−no−fork
Don´t try to fork the repository.

−−mirror
Use the specified *URL* as a mirror URL. If *URL* is a comma−separated list of URLs, multiple mirrors will be added.

−−fork−org
Use the specified GitHub organization for forking.

−−version
Use the specified *version* to override the value parsed from the URL or tag. Note that **−−version=0** can be used to delete an existing version override from a formula if it has become redundant.

−−message
Append *message* to the default pull request message.

−−url Specify the *URL* for the new download. If a *URL* is specified, the *SHA−256* checksum of the new download should also be specified.

−−sha256
Specify the *SHA−256* checksum of the new download.

−−tag Specify the new git commit *tag* for the formula.

−−revision
Specify the new commit *revision* corresponding to the specified git *tag* or specified *version*.

−f, **−−force**
Ignore duplicate open PRs. Remove all mirrors if **−−mirror** was not specified.

bump−revision [*options*] *formula* [...]
Create a commit to increment the revision of *formula*. If no revision is present, "revision 1" will be added.

−n, **−−dry−run**
Print what would be done rather than doing it.

−−remove−bottle−block
Remove the bottle block in addition to bumping the revision.

−−write−only
Make the expected file modifications without taking any Git actions.

−−message
Append *message* to the default commit message.

bump−unversioned−casks [*options*] *cask*|*tap* [...]
Check all casks with unversioned URLs in a given *tap* for updates.

−n, **−−dry−run**
Do everything except caching state and opening pull requests.

−−limit
Maximum runtime in minutes.

−−state−file
File for caching state.

cat [−−*formula*] [−−*cask*] *formula*|*cask*
Display the source of a *formula* or *cask*.

−−formula
Treat all named arguments as formulae.

−−cask Treat all named arguments as casks.

command *command* [...]
Display the path to the file being used when invoking **brew** *cmd*.

create [*options*] *URL*
Generate a formula or, with **−−cask**, a cask for the downloadable file at *URL* and open it in the editor. Homebrew will attempt to automatically derive the formula name and version, but if it fails, you´ll have to make your own template. The **wget** formula serves as a simple example. For the complete API, see: *https://rubydoc.brew.sh/Formula*

−−autotools
Create a basic template for an Autotools−style build.

−−cask Create a basic template for a cask.

−−cmake
Create a basic template for a CMake−style build.

−−crystal
Create a basic template for a Crystal build.

−−go Create a basic template for a Go build.

−−meson
Create a basic template for a Meson−style build.

−−node
Create a basic template for a Node build.

−−perl Create a basic template for a Perl build.

−−python
Create a basic template for a Python build.

−−ruby
Create a basic template for a Ruby build.

−−rust Create a basic template for a Rust build.

−−no−fetch
Homebrew will not download *URL* to the cache and will thus not add its SHA−256 to the formula for you, nor will it check the GitHub API for GitHub projects (to fill out its description and homepage).

−−HEAD
Indicate that *URL* points to the package´s repository rather than a file.

−−set−name
Explicitly set the *name* of the new formula or cask.

−−set−version
Explicitly set the *version* of the new formula or cask.

−−set−license
Explicitly set the *license* of the new formula.

−−tap Generate the new formula within the given tap, specified as *user/repo*.

−f, **−−force**
Ignore errors for disallowed formula names and names that shadow aliases.

dispatch−build−bottle [*options*] *formula* [...]
Build bottles for these formulae with GitHub Actions.

−−tap Target tap repository (default: **homebrew/core**).

−−timeout
Build timeout (in minutes, default: 60).

−−issue
If specified, post a comment to this issue number if the job fails.

−−macos
Version(s) of macOS the bottle should be built for.

−−workflow
Dispatch specified workflow (default: **dispatch−build−bottle.yml**).

−−upload
Upload built bottles.

−−linux
Dispatch bottle for Linux (using GitHub runners).

−−linux−self−hosted
Dispatch bottle for Linux (using self−hosted runner).

−−linux−wheezy
Use Debian Wheezy container for building the bottle on Linux.

edit [*options*] [*formula*|*cask* ...]
Open a *formula* or *cask* in the editor set by **EDITOR** or **HOMEBREW_EDITOR**, or open the Homebrew repository for editing if no formula is provided.

−−formula
Treat all named arguments as formulae.

−−cask Treat all named arguments as casks.

−−print−path
Print the file path to be edited, without opening an editor.

extract [*−−version*=] [*−−force*] *formula tap*
Look through repository history to find the most recent version of *formula* and create a copy in *tap*. Specifically, the command will create the new formula file at *tap*/**Formula**/*formula*@*version*.**rb**. If the tap is not installed yet, attempt to install/clone the tap before continuing. To extract a formula from a tap that is not **homebrew/core** use its fully−qualified form of *user*/*repo*/*formula*.

−−version
Extract the specified *version* of *formula* instead of the most recent.

−f, **−−force**
Overwrite the destination formula if it already exists.

formula *formula* [...]
Display the path where *formula* is located.

generate−man−completions [*−−fail−if−not−changed*]
Generate Homebrew´s manpages and shell completions.

−−fail−if−not−changed
Return a failing status code if no changes are detected in the manpage outputs. This can be used to notify CI when the manpages are out of date. Additionally, the date used in new manpages will match those in the existing manpages (to allow comparison without factoring in the date).

install−bundler−gems [−−*groups*=]
Install Homebrew´s Bundler gems.

−−groups
Installs the specified comma−separated list of gem groups (default: last used).

irb [−−*examples*] [−−*pry*]
Enter the interactive Homebrew Ruby shell.

−−examples
Show several examples.

−−pry Use Pry instead of IRB. Implied if **HOMEBREW_PRY** is set.

linkage [*options*] [*installed_formula* ...]
Check the library links from the given *formula* kegs. If no *formula* are provided, check all kegs. Raises an error if run on uninstalled formulae.

−−test Show only missing libraries and exit with a non−zero status if any missing libraries are found.

−−strict
Exit with a non−zero status if any undeclared dependencies with linkage are found.

−−reverse
For every library that a keg references, print its dylib path followed by the binaries that link to it.

−−cached
Print the cached linkage values stored in **HOMEBREW_CACHE**, set by a previous **brew linkage** run.

livecheck, **lc** [*options*] [*formula*|*cask* ...]
Check for newer versions of formulae and/or casks from upstream.

If no formula or cask argument is passed, the list of formulae and casks to check is taken from **HOMEBREW_LIVECHECK_WATCHLIST** or **˜/.brew_livecheck_watchlist**.

−−full−name
Print formulae/casks with fully−qualified names.

−−tap Check formulae/casks within the given tap, specified as *user*/*repo*.

−−all Check all available formulae/casks.

−−installed
Check formulae/casks that are currently installed.

−−newer−only
Show the latest version only if it´s newer than the formula/cask.

−−json Output information in JSON format.

−q, **−−quiet**
Suppress warnings, don´t print a progress bar for JSON output.

−−formula
Only check formulae.

−−cask Only check casks.

pr−automerge [*options*]
Find pull requests that can be automatically merged using **brew pr−publish**.

−−tap Target tap repository (default: **homebrew/core**).

−−with−label
Pull requests must have this label.

−−without−labels
Pull requests must not have these labels (default: **do not merge**, **new formula**, **automerge−skip**).

−−without−approval
Pull requests do not require approval to be merged.

−−publish
Run **brew pr−publish** on matching pull requests.

−−no−autosquash
Instruct **brew pr−publish** to skip automatically reformatting and rewording commits in the pull request to the preferred format.

−−ignore−failures
Include pull requests that have failing status checks.

pr−publish [*options*] *pull_request* [...]
Publish bottles for a pull request with GitHub Actions. Requires write access to the repository.

−−no−autosquash
Skip automatically reformatting and rewording commits in the pull request to the preferred format, even if supported on the target tap.

−−branch
Branch to publish to (default: **master**).

−−message
Message to include when autosquashing revision bumps, deletions, and rebuilds.

−−tap Target tap repository (default: **homebrew/core**).

−−workflow
Target workflow filename (default: **publish−commit−bottles.yml**).

pr−pull [*options*] *pull_request* [...]
Download and publish bottles, and apply the bottle commit from a pull request with artifacts generated by GitHub Actions. Requires write access to the repository.

−−no−upload
Download the bottles but don´t upload them.

−−no−commit
Do not generate a new commit before uploading.

−n, **−−dry−run**
Print what would be done rather than doing it.

−−clean
Do not amend the commits from pull requests.

−−keep−old
If the formula specifies a rebuild version, attempt to preserve its value in the generated DSL.

−−no−autosquash
Skip automatically reformatting and rewording commits in the pull request to our preferred format.

−−branch−okay
Do not warn if pulling to a branch besides the repository default (useful for testing).

−−resolve
When a patch fails to apply, leave in progress and allow user to resolve, instead of aborting.

−−warn−on−upload−failure
Warn instead of raising an error if the bottle upload fails. Useful for repairing bottle uploads that previously failed.

−−committer
Specify a committer name and email in **git**´s standard author format.

−−message
Message to include when autosquashing revision bumps, deletions, and rebuilds.

−−artifact
Download artifacts with the specified name (default: **bottles**).

−−tap Target tap repository (default: **homebrew/core**).

−−root−url
Use the specified *URL* as the root of the bottle´s URL instead of Homebrew´s default.

−−root−url−using
Use the specified download strategy class for downloading the bottle´s URL instead of Homebrew´s default.

−−workflows
Retrieve artifacts from the specified workflow (default: **tests.yml**). Can be a comma−separated list to include multiple workflows.

−−ignore−missing−artifacts
Comma−separated list of workflows which can be ignored if they have not been run.

pr−upload [*options*]
Apply the bottle commit and publish bottles to a host.

−−keep−old
If the formula specifies a rebuild version, attempt to preserve its value in the generated DSL.

−n, **−−dry−run**
Print what would be done rather than doing it.

−−no−commit
Do not generate a new commit before uploading.

−−warn−on−upload−failure
Warn instead of raising an error if the bottle upload fails. Useful for repairing bottle uploads that previously failed.

−−upload−only
Skip running **brew bottle** before uploading.

−−committer
Specify a committer name and email in **git**´s standard author format.

−−root−url
Use the specified *URL* as the root of the bottle´s URL instead of Homebrew´s default.

−−root−url−using
Use the specified download strategy class for downloading the bottle´s URL instead of Homebrew´s default.

prof [−−*stackprof*] *command* [...]
Run Homebrew with a Ruby profiler. For example, **brew prof readall**.

−−stackprof
Use **stackprof** instead of **ruby−prof** (the default).

release [−−*major*] [−−*minor*]
Create a new draft Homebrew/brew release with the appropriate version number and release notes.

By default, **brew release** will bump the patch version number. Pass **−−major** or **−−minor** to bump the major or minor version numbers, respectively. The command will fail if the previous major or minor release was made less than one month ago.

Requires write access to the Homebrew/brew repository.

−−major
Create a major release.

−−minor
Create a minor release.

rubocop
Installs, configures and runs Homebrew´s **rubocop**.

ruby [*options*] (**−e** *text*|*file*)
Run a Ruby instance with Homebrew´s libraries loaded. For example, **brew ruby −e "puts :gcc.f.deps"** or **brew ruby script.rb**.

−r Load a library using **require**.

−e Execute the given text string as a script.

sh [−−*env*=] [−−*cmd*=] [*file*]
Enter an interactive shell for Homebrew´s build environment. Use years−battle−hardened build logic to help your **./configure && make && make install** and even your **gem install** succeed. Especially handy if you run Homebrew in an Xcode−only configuration since it adds tools like **make** to your **PATH** which build systems would not find otherwise.

−−env Use the standard **PATH** instead of superenv´s when **std** is passed.

−c, **−−cmd**
Execute commands in a non−interactive shell.

sponsors
Update the list of GitHub Sponsors in the **Homebrew/brew** README.

style [*options*] [*file*|*tap*|*formula*|*cask* ...]
Check formulae or files for conformance to Homebrew style guidelines.

Lists of *file*, *tap* and *formula* may not be combined. If none are provided, **style** will run style checks on the whole Homebrew library, including core code and all formulae.

−−fix Fix style violations automatically using RuboCop´s auto−correct feature.

−−display−cop−names
Include the RuboCop cop name for each violation in the output.

−−reset−cache
Reset the RuboCop cache.

−−formula
Treat all named arguments as formulae.

−−cask Treat all named arguments as casks.

−−only−cops
Specify a comma−separated *cops* list to check for violations of only the listed RuboCop cops.

−−except−cops
Specify a comma−separated *cops* list to skip checking for violations of the listed RuboCop cops.

tap−new [*options*] *user*/*repo*
Generate the template files for a new tap.

−−no−git
Don´t initialize a Git repository for the tap.

−−pull−label
Label name for pull requests ready to be pulled (default: **pr−pull**).

−−branch
Initialize Git repository and setup GitHub Actions workflows with the specified branch name (default: **main**).

−−github−packages
Upload bottles to GitHub Packages.

test [*options*] *installed_formula* [...]
Run the test method provided by an installed formula. There is no standard output or return code, but generally it should notify the user if something is wrong with the installed formula.

Example: **brew install jruby && brew test jruby**

−f, **−−force**
Test formulae even if they are unlinked.

−−HEAD
Test the head version of a formula.

−−keep−tmp
Retain the temporary files created for the test.

−−retry
Retry if a testing fails.

tests [*options*]
Run Homebrew´s unit and integration tests.

−−coverage
Generate code coverage reports.

−−generic
Run only OS−agnostic tests.

−−no−compat
Do not load the compatibility layer when running tests.

−−online
Include tests that use the GitHub API and tests that use any of the taps for official external commands.

−−byebug
Enable debugging using byebug.

−−only Run only *test_script*_**spec.rb**. Appending **:***line_number* will start at a specific line.

−−seed Randomise tests with the specified *value* instead of a random seed.

typecheck, **tc** [*options*]
Check for typechecking errors using Sorbet.

−−fix Automatically fix type errors.

−q, **−−quiet**
Silence all non−critical errors.

−−update
Update RBI files.

−−suggest−typed
Try upgrading **typed** sigils.

−−fail−if−not−changed
Return a failing status code if all gems are up to date and gem definitions do not need a tapioca update.

−−dir Typecheck all files in a specific directory.

−−file Typecheck a single file.

−−ignore
Ignores input files that contain the given string in their paths (relative to the input path passed to Sorbet).

unbottled [*options*] [*formula* ...]
Show the unbottled dependents of formulae.

−−tag Use the specified bottle tag (e.g. **big_sur**) instead of the current OS.

−−dependents
Skip getting analytics data and sort by number of dependents instead.

−−all Print the number of unbottled and total formulae.

unpack [*options*] *formula* [...]
Unpack the source files for *formula* into subdirectories of the current working directory.

−−destdir
Create subdirectories in the directory named by *path* instead.

−−patch
Patches for *formula* will be applied to the unpacked source.

−g, **−−git**
Initialise a Git repository in the unpacked source. This is useful for creating patches for the software.

−f, **−−force**
Overwrite the destination directory if it already exists.

update−license−data [*−−fail−if−not−changed*]
Update SPDX license data in the Homebrew repository.

−−fail−if−not−changed
Return a failing status code if current license data´s version is the same as the upstream. This can be used to notify CI when the SPDX license data is out of date.

update−maintainers
Update the list of maintainers in the **Homebrew/brew** README.

update−python−resources [*options*] *formula* [...]
Update versions for PyPI resource blocks in *formula*.

−p, **−−print−only**
Print the updated resource blocks instead of changing *formula*.

−s, **−−silent**
Suppress any output.

−−ignore−non−pypi−packages
Don´t fail if *formula* is not a PyPI package.

−−version
Use the specified *version* when finding resources for *formula*. If no version is specified, the current version for *formula* will be used.

−−package−name
Use the specified *package−name* when finding resources for *formula*. If no package name is specified, it will be inferred from the formula´s stable URL.

−−extra−packages
Include these additional packages when finding resources.

−−exclude−packages
Exclude these packages when finding resources.

update−test [*options*]
Run a test of **brew update** with a new repository clone. If no options are passed, use **origin/master** as the start commit.

−−to−tag
Set **HOMEBREW_UPDATE_TO_TAG** to test updating between tags.

−−keep−tmp
Retain the temporary directory containing the new repository clone.

−−commit
Use the specified *commit* as the start commit.

−−before
Use the commit at the specified *date* as the start commit.

vendor−gems [−−*update*=]
Install and commit Homebrew´s vendored gems.

−−update
Update all vendored Gems to the latest version.

GLOBAL CASK OPTIONS

These options are applicable to the **install**, **reinstall**, and **upgrade** subcommands with the **−−cask** flag.

−−appdir
Target location for Applications (default: **/Applications**).

−−colorpickerdir
Target location for Color Pickers (default: **˜/Library/ColorPickers**).

−−prefpanedir
Target location for Preference Panes (default: **˜/Library/PreferencePanes**).

−−qlplugindir
Target location for QuickLook Plugins (default: **˜/Library/QuickLook**).

−−mdimporterdir
Target location for Spotlight Plugins (default: **˜/Library/Spotlight**).

−−dictionarydir
Target location for Dictionaries (default: **˜/Library/Dictionaries**).

−−fontdir
Target location for Fonts (default: **˜/Library/Fonts**).

−−servicedir
Target location for Services (default: **˜/Library/Services**).

−−input−methoddir
Target location for Input Methods (default: **˜/Library/Input Methods**).

−−internet−plugindir
Target location for Internet Plugins (default: **˜/Library/Internet Plug−Ins**).

−−audio−unit−plugindir
Target location for Audio Unit Plugins (default: **˜/Library/Audio/Plug−Ins/Components**).

−−vst−plugindir
Target location for VST Plugins (default: **˜/Library/Audio/Plug−Ins/VST**).

−−vst3−plugindir
Target location for VST3 Plugins (default: **˜/Library/Audio/Plug−Ins/VST3**).

−−screen−saverdir
Target location for Screen Savers (default: **˜/Library/Screen Savers**).

−−language
Comma−separated list of language codes to prefer for cask installation. The first matching language is used, otherwise it reverts to the cask´s default language. The default value is the language of your system.

GLOBAL OPTIONS

These options are applicable across multiple subcommands.

−d, **−−debug**
Display any debugging information.

−q, **−−quiet**
Make some output more quiet.

−v, **−−verbose**
Make some output more verbose.

−h, **−−help**
Show this message.

OFFICIAL EXTERNAL COMMANDS

alias [*alias* ... | *alias*=*command*]
Show existing aliases. If no aliases are given, print the whole list.

−−edit Edit aliases in a text editor. Either one or all aliases may be opened at once. If the given alias doesn´t exist it´ll be pre−populated with a template.

autoupdate *subcommand* [*interval*] [*options*]
An easy, convenient way to automatically update Homebrew.

This script will run **brew update** in the background once every 24 hours (by default) until explicitly told to stop, utilising **launchd**.

brew autoupdate start [**interval**] [**options**]
Start autoupdating either once every **interval** hours or once every 24 hours. Please note the interval has to be passed in seconds, so 12 hours would be **brew autoupdate start 43200**. Pass **−−upgrade** or **−−cleanup** to automatically run **brew upgrade** and/or **brew cleanup** respectively. Pass **−−enable−notification** to send a notification when the autoupdate process has finished successfully.

brew autoupdate stop
Stop autoupdating, but retain plist & logs.

brew autoupdate delete
Cancel the autoupdate, delete the plist and logs.

brew autoupdate status
Prints the current status of this tool.

brew autoupdate version
Output this tool´s current version, and a short changelog.

−−upgrade
Automatically upgrade your installed formulae. If the Caskroom exists locally Casks will be upgraded as well. Must be passed with **start**.

−−greedy
Upgrade casks with −−greedy (include auto−updating casks). Must be passed with **start**.

−−cleanup
Automatically clean brew´s cache and logs. Must be passed with **start**.

−−enable−notification
Send a notification when the autoupdate process has finished successfully, if **terminal−notifier** is installed & found. Must be passed with **start**. <NOTE: Notifications are enabled by default on macOS Catalina and newer.>

−−immediate
Starts the autoupdate command immediately, instead of waiting for one interval (24 hours by default) to pass first. Must be passed with **start**.

bundle [*subcommand*]
Bundler for non−Ruby dependencies from Homebrew, Homebrew Cask, Mac App Store and Whalebrew.

brew bundle [**install**]
Install and upgrade (by default) all dependencies from the **Brewfile**.

You can specify the **Brewfile** location using **−−file** or by setting the **HOMEBREW_BUNDLE_FILE** environment variable.

You can skip the installation of dependencies by adding space−separated values to one or more of the following environment variables: **HOMEBREW_BUNDLE_BREW_SKIP**, **HOMEBREW_BUNDLE_CASK_SKIP**, **HOMEBREW_BUNDLE_MAS_SKIP**, **HOMEBREW_BUNDLE_WHALEBREW_SKIP**, **HOMEBREW_BUNDLE_TAP_SKIP**.

brew bundle will output a **Brewfile.lock.json** in the same directory as the **Brewfile** if all dependencies are installed successfully. This contains dependency and system status information which can be useful in debugging **brew bundle** failures and replicating a "last known good build" state. You can opt−out of this behaviour by setting the **HOMEBREW_BUNDLE_NO_LOCK** environment variable or passing the **−−no−lock** option. You may wish to check this file into the same version control system as your **Brewfile** (or ensure your version control system ignores it if you´d prefer to rely on debugging information from a local machine).

brew bundle dump
Write all installed casks/formulae/images/taps into a **Brewfile** in the current directory.

brew bundle cleanup
Uninstall all dependencies not listed from the **Brewfile**.

This workflow is useful for maintainers or testers who regularly install lots of formulae.

brew bundle check
Check if all dependencies are installed from the **Brewfile**.

This provides a successful exit code if everything is up−to−date, making it useful for scripting.

brew bundle list
List all dependencies present in the **Brewfile**.

By default, only Homebrew dependencies are listed.

brew bundle exec *command*
Run an external command in an isolated build environment based on the **Brewfile** dependencies.

This sanitized build environment ignores unrequested dependencies, which makes sure that things you didn´t specify in your **Brewfile** won´t get picked up by commands like **bundle install**, **npm install**, etc. It will also add compiler flags which will help find keg−only dependencies like **openssl**, **icu4c**, etc.

−−file Read the **Brewfile** from this location. Use **−−file=−** to pipe to stdin/stdout.

−−global
Read the **Brewfile** from **~/.Brewfile**.

−v, **−−verbose**
install prints output from commands as they are run. **check** lists all missing dependencies.

−−no−upgrade
: **install** won´t run **brew upgrade** on outdated dependencies. Note they may still be upgraded by **brew install** if needed.

−f, **−−force**
: **dump** overwrites an existing **Brewfile**. **cleanup** actually performs its cleanup operations.

−−cleanup
: **install** performs cleanup operation, same as running **cleanup −−force**.

−−no−lock
: **install** won´t output a **Brewfile.lock.json**.

−−all **list** all dependencies.

−−formula
: **list** Homebrew dependencies.

−−cask **list** Homebrew Cask dependencies.

−−tap **list** tap dependencies.

−−mas **list** Mac App Store dependencies.

−−whalebrew
: **list** Whalebrew dependencies.

−−describe
: **dump** adds a description comment above each line, unless the dependency does not have a description.

−−no−restart
: **dump** does not add **restart_service** to formula lines.

−−zap **cleanup** casks using the **zap** command instead of **uninstall**.

command−not−found−init
Print instructions for setting up the command−not−found hook for your shell. If the output is not to a tty, print the appropriate handler script for your shell.

services [*subcommand*]
Manage background services with macOS´ **launchctl**(1) daemon manager.

If **sudo** is passed, operate on **/Library/LaunchDaemons** (started at boot). Otherwise, operate on **˜/Library/LaunchAgents** (started at login).

[**sudo**] **brew services** [**list**] (**−−json**)
List information about all managed services for the current user (or root).

[**sudo**] **brew services info** (*formula*|**−−all**|**−−json**)
List all managed services for the current user (or root).

[**sudo**] **brew services run** (*formula*|**−−all**)
Run the service *formula* without registering to launch at login (or boot).

[**sudo**] **brew services start** (*formula*|**−−all**)
Start the service *formula* immediately and register it to launch at login (or boot).

[**sudo**] **brew services stop** (*formula*|**−−all**)
Stop the service *formula* immediately and unregister it from launching at login (or boot).

[**sudo**] **brew services kill** (*formula*|**−−all**)
Stop the service *formula* immediately but keep it registered to launch at login (or boot).

[**sudo**] **brew services restart** (*formula*|**−−all**)
Stop (if necessary) and start the service *formula* immediately and register it to launch at login (or boot).

[**sudo**] **brew services cleanup**

Remove all unused services.

−−file Use the plist file from this location to **start** or **run** the service.

−−all Run *subcommand* on all services.

−−json Output as JSON.

test−bot [*options*] [*formula*]
Tests the full lifecycle of a Homebrew change to a tap (Git repository). For example, for a GitHub Actions pull request that changes a formula **brew test−bot** will ensure the system is cleaned and set up to test the formula, install the formula, run various tests and checks on it, bottle (package) the binaries and test formulae that depend on it to ensure they aren´t broken by these changes.

Only supports GitHub Actions as a CI provider. This is because Homebrew uses GitHub Actions and it´s freely available for public and private use with macOS and Linux workers.

−−dry−run
Print what would be done rather than doing it.

−−cleanup
Clean all state from the Homebrew directory. Use with care!

−−skip−setup
Don´t check if the local system is set up correctly.

−−build−from−source
Build from source rather than building bottles.

−−build−dependents−from−source
Build dependents from source rather than testing bottles.

−−junit
generate a JUnit XML test results file.

−−keep−old
Run **brew bottle −−keep−old** to build new bottles for a single platform.

−−skip−relocation
Run **brew bottle −−skip−relocation** to build new bottles that don´t require relocation.

−−only−json−tab
Run **brew bottle −−only−json−tab** to build new bottles that do not contain a tab.

−−local
Ask Homebrew to write verbose logs under **./logs/** and set **$HOME** to **./home/**

−−tap Use the Git repository of the given tap. Defaults to the core tap for syntax checking.

−−fail−fast
Immediately exit on a failing step.

−v, **−−verbose**
Print test step output in real time. Has the side effect of passing output as raw bytes instead of re−encoding in UTF−8.

−−test−default−formula
Use a default testing formula when not building a tap and no other formulae are specified.

−−root−url
Use the specified *URL* as the root of the bottle´s URL instead of Homebrew´s default.

−−git−name
Set the Git author/committer names to the given name.

−−git−email
Set the Git author/committer email to the given email.

−−publish
Publish the uploaded bottles.

−−skip−online−checks
Don´t pass **−−online** to **brew audit** and skip **brew livecheck**.

−−skip−dependents
Don´t test any dependents.

−−skip−recursive−dependents
Only test the direct dependents.

−−only−cleanup−before
Only run the pre−cleanup step. Needs **−−cleanup**.

−−only−setup
Only run the local system setup check step.

−−only−tap−syntax
Only run the tap syntax check step.

−−only−formulae
Only run the formulae steps.

−−only−formulae−detect
Only run the formulae detection steps.

−−only−formulae−dependents
Only run the formulae dependents steps.

−−only−cleanup−after
Only run the post−cleanup step. Needs **−−cleanup**.

−−testing−formulae
Use these testing formulae rather than running the formulae detection steps.

−−added−formulae
Use these added formulae rather than running the formulae detection steps.

−−deleted−formulae
Use these deleted formulae rather than running the formulae detection steps.

−−skipped−or−failed−formulae
Use these skipped or failed formulae from formulae steps for a formulae dependents step.

unalias *alias* [...]
Remove aliases.

which−formula [*−−explain*] *command* [...]
Prints the formula(e) which provides the given command.

−−explain
Output explanation of how to get ´cmd´ by installing one of the providing formulae.

which−update [*options*] [*database*]
Database update for **brew which−formula**

−−stats Print statistics about the database contents (number of commands and formulae, list of missing formulae).

−−commit
Commit the changes using **git**.

−−update−existing
Update database entries with outdated formula versions.

−−install−missing
Install and update formulae that are missing from the database and don´t have bottles.

−−max−downloads
Specify a maximum number of formulae to download and update.

CUSTOM EXTERNAL COMMANDS

Homebrew, like **git**(1), supports external commands. These are executable scripts that reside somewhere in the **PATH**, named **brew−***cmdname* or **brew−***cmdname***.rb**, which can be invoked like **brew** *cmdname*. This allows you to create your own commands without modifying Homebrew´s internals.

Instructions for creating your own commands can be found in the docs: *https://docs.brew.sh/External−Commands*

SPECIFYING FORMULAE

Many Homebrew commands accept one or more *formula* arguments. These arguments can take several different forms:

The name of a formula
e.g. **git**, **node**, **wget**.

The fully−qualified name of a tapped formula
Sometimes a formula from a tapped repository may conflict with one in **homebrew/core**. You can still access these formulae by using a special syntax, e.g. **homebrew/dupes/vim** or **homebrew/versions/node4**.

An arbitrary file
Homebrew can install formulae from a local path. It can point to either a formula file or a bottle. Prefix relative paths with **./** to prevent them from being interpreted as a formula or tap name.

SPECIFYING CASKS

Many Homebrew Cask commands accept one or more *cask* arguments. These can be specified the same way as the *formula* arguments described in **SPECIFYING FORMULAE** above.

ENVIRONMENT

Note that environment variables must have a value set to be detected. For example, run **export HOMEBREW_NO_INSECURE_REDIRECT=1** rather than just **export HOMEBREW_NO_INSECURE_REDIRECT**.

HOMEBREW_ADDITIONAL_GOOGLE_ANALYTICS_ID
Additional Google Analytics tracking ID to emit user behaviour analytics to. For more information, see: *https://docs.brew.sh/Analytics*

HOMEBREW_ARCH
Linux only: Pass this value to a type name representing the compiler´s **−march** option.

Default: **native**.

HOMEBREW_ARTIFACT_DOMAIN
Prefix all download URLs, including those for bottles, with this value. For example, **HOMEBREW_ARTIFACT_DOMAIN=http://localhost:8080** will cause a formula with the URL **https://example.com/foo.tar.gz** to instead download from **http://localhost:8080/example.com/foo.tar.gz**.

HOMEBREW_AUTO_UPDATE_SECS
Run **brew update** once every **HOMEBREW_AUTO_UPDATE_SECS** seconds before some commands, e.g. **brew install**, **brew upgrade** and **brew tap**. Alternatively, disable auto−update entirely with HOMEBREW_NO_AUTO_UPDATE.

Default: **300**.

HOMEBREW_BAT
If set, use **bat** for the **brew cat** command.

HOMEBREW_BAT_CONFIG_PATH
Use this as the **bat** configuration file.

Default: **$HOME/.config/bat/config**.

HOMEBREW_BOOTSNAP
If set, use Bootsnap to speed up repeated **brew** calls. A no−op when using Homebrew´s vendored, relocatable Ruby on macOS (as it doesn´t work).

HOMEBREW_BOTTLE_DOMAIN
Use this URL as the download mirror for bottles. If bottles at that URL are temporarily unavailable, the default bottle domain will be used as a fallback mirror. For example, **HOMEBREW_BOTTLE_DOMAIN=http://localhost:8080** will cause all bottles to download from the prefix **http://localhost:8080/**. If bottles are not available at **HOMEBREW_BOTTLE_DOMAIN** they will be downloaded from the default bottle domain.

Default: **https://ghcr.io/v2/homebrew/core**.

HOMEBREW_BREW_GIT_REMOTE
Use this URL as the Homebrew/brew **git**(1) remote.

Default: **https://github.com/Homebrew/brew**.

HOMEBREW_BROWSER
Use this as the browser when opening project homepages.

Default: **$BROWSER** or the OS´s default browser.

HOMEBREW_CACHE
Use this directory as the download cache.

Default: macOS: **$HOME/Library/Caches/Homebrew**, Linux: **$XDG_CACHE_HOME/Homebrew** or **$HOME/.cache/Homebrew**.

HOMEBREW_CASK_OPTS
Append these options to all **cask** commands. All **−−*dir** options, **−−language**, **−−require−sha**, **−−no−quarantine** and **−−no−binaries** are supported. For example, you might add something like the following to your **˜/.profile**, **˜/.bash_profile**, or **˜/.zshenv**:

export HOMEBREW_CASK_OPTS="−−appdir=˜/Applications −−fontdir=/Library/Fonts"

HOMEBREW_CLEANUP_PERIODIC_FULL_DAYS
If set, **brew install**, **brew upgrade** and **brew reinstall** will cleanup all formulae when this number of days has passed.

Default: **30**.

HOMEBREW_CLEANUP_MAX_AGE_DAYS
Cleanup all cached files older than this many days.

Default: **120**.

HOMEBREW_COLOR
If set, force colour output on non−TTY outputs.

HOMEBREW_CORE_GIT_REMOTE
Use this URL as the Homebrew/homebrew−core **git**(1) remote.

Default: **https://github.com/Homebrew/homebrew−core**.

HOMEBREW_CURLRC
If set, do not pass **−−disable** when invoking **curl**(1), which disables the use of **curlrc**.

HOMEBREW_CURL_RETRIES
Pass the given retry count to **−−retry** when invoking **curl**(1).

Default: **3**.

HOMEBREW_CURL_VERBOSE
If set, pass **−−verbose** when invoking **curl**(1).

HOMEBREW_DEVELOPER
If set, tweak behaviour to be more relevant for Homebrew developers (active or budding) by e.g. turning warnings into errors.

HOMEBREW_DISABLE_LOAD_FORMULA
If set, refuse to load formulae. This is useful when formulae are not trusted (such as in pull requests).

HOMEBREW_DISPLAY
Use this X11 display when opening a page in a browser, for example with **brew home**. Primarily useful on Linux.

Default: **$DISPLAY**.

HOMEBREW_DISPLAY_INSTALL_TIMES
If set, print install times for each formula at the end of the run.

HOMEBREW_EDITOR
Use this editor when editing a single formula, or several formulae in the same directory.

Note: **brew edit** will open all of Homebrew as discontinuous files and directories. Visual Studio Code can handle this correctly in project mode, but many editors will do strange things in this case.

Default: **$EDITOR** or **$VISUAL**.

HOMEBREW_FAIL_LOG_LINES
Output this many lines of output on formula **system** failures.

Default: **15**.

HOMEBREW_FORBIDDEN_LICENSES
A space−separated list of licenses. Homebrew will refuse to install a formula if it or any of its dependencies has a license on this list.

HOMEBREW_FORCE_BREWED_CA_CERTIFICATES
If set, always use a Homebrew−installed **ca−certificates** rather than the system version. Automatically set if the system version is too old.

HOMEBREW_FORCE_BREWED_CURL
If set, always use a Homebrew−installed **curl**(1) rather than the system version. Automatically set if the system version of **curl** is too old.

HOMEBREW_FORCE_BREWED_GIT
If set, always use a Homebrew−installed **git**(1) rather than the system version. Automatically set if the system version of **git** is too old.

HOMEBREW_FORCE_VENDOR_RUBY
If set, always use Homebrew´s vendored, relocatable Ruby version even if the system version of Ruby is new enough.

HOMEBREW_GITHUB_API_TOKEN
Use this personal access token for the GitHub API, for features such as **brew search**. You can create one at *https://github.com/settings/tokens*. If set, GitHub will allow you a greater number of API requests. For more information, see: *https://docs.github.com/en/rest/overview/resources−in−the−rest−api#rate−limiting*

Note: Homebrew doesn´t require permissions for any of the scopes, but some developer commands may require additional permissions.

HOMEBREW_GITHUB_PACKAGES_TOKEN
Use this GitHub personal access token when accessing the GitHub Packages Registry (where bottles may be stored).

HOMEBREW_DOCKER_REGISTRY_BASIC_AUTH_TOKEN
Use this base64 encoded username and password for authenticating with a Docker registry proxying GitHub Packages. If HOMEBREW_DOCKER_REGISTRY_TOKEN is set, it will be used instead.

HOMEBREW_DOCKER_REGISTRY_TOKEN
Use this bearer token for authenticating with a Docker registry proxying GitHub Packages. Preferred over HOMEBREW_DOCKER_REGISTRY_TOKEN_BASIC.

HOMEBREW_GITHUB_PACKAGES_USER
Use this username when accessing the GitHub Packages Registry (where bottles may be stored).

HOMEBREW_GIT_EMAIL
Set the Git author and committer email to this value.

HOMEBREW_GIT_NAME
Set the Git author and committer name to this value.

HOMEBREW_INSTALL_BADGE
Print this text before the installation summary of each successful build.

Default: The "Beer Mug" emoji.

HOMEBREW_INSTALL_FROM_API
If set, install formulae and casks in homebrew/core and homebrew/cask taps using Homebrew´s API instead of needing (large, slow) local checkouts of these repositories.

Note: Setting HOMEBREW_INSTALL_FROM_API is not compatible with Homebrew´s developer mode so will error (as Homebrew development needs a full clone).

HOMEBREW_LIVECHECK_WATCHLIST
Consult this file for the list of formulae to check by default when no formula argument is passed to **brew livecheck**.

Default: **$HOME/.brew_livecheck_watchlist**.

HOMEBREW_LOGS
Use this directory to store log files.

Default: macOS: **$HOME/Library/Logs/Homebrew**, Linux: **$XDG_CACHE_HOME/Homebrew/Logs** or **$HOME/.cache/Homebrew/Logs**.

HOMEBREW_MAKE_JOBS
Use this value as the number of parallel jobs to run when building with **make**(1).

Default: The number of available CPU cores.

HOMEBREW_NO_ANALYTICS
If set, do not send analytics. For more information, see: *https://docs.brew.sh/Analytics*

HOMEBREW_NO_AUTO_UPDATE
If set, do not automatically update before running some commands, e.g. **brew install**, **brew upgrade** and **brew tap**. Alternatively, run this less often by setting HOMEBREW_AUTO_UPDATE_SECS to a value higher than the default.

HOMEBREW_NO_BOOTSNAP
If set, do not use Bootsnap to speed up repeated **brew** calls.

HOMEBREW_NO_INSTALLED_DEPENDENTS_CHECK
If set, do not check for broken linkage of dependents or outdated dependents after installing, upgrading or reinstalling formulae. This will result in fewer dependents (and their dependencies) being upgraded or reinstalled but may result in more breakage from running **brew install**

<formula> or **brew upgrade <formula>**.

HOMEBREW_NO_CLEANUP_FORMULAE
A comma−separated list of formulae. Homebrew will refuse to clean up a formula if it appears on this list.

HOMEBREW_NO_COLOR
If set, do not print text with colour added.

Default: **$NO_COLOR**.

HOMEBREW_NO_COMPAT
If set, disable all use of legacy compatibility code.

HOMEBREW_NO_EMOJI
If set, do not print **HOMEBREW_INSTALL_BADGE** on a successful build.

Note: Will only try to print emoji on OS X Lion or newer.

HOMEBREW_NO_ENV_HINTS
If set, do not print any hints about changing Homebrew´s behaviour with environment variables.

HOMEBREW_NO_GITHUB_API
If set, do not use the GitHub API, e.g. for searches or fetching relevant issues after a failed install.

HOMEBREW_NO_INSECURE_REDIRECT
If set, forbid redirects from secure HTTPS to insecure HTTP.

Note: While ensuring your downloads are fully secure, this is likely to cause from−source Source-Forge, some GNU & GNOME−hosted formulae to fail to download.

HOMEBREW_NO_INSTALL_CLEANUP
If set, **brew install**, **brew upgrade** and **brew reinstall** will never automatically cleanup installed/upgraded/reinstalled formulae or all formulae every **HOMEBREW_CLEANUP_PERIODIC_FULL_DAYS** days. Alternatively, HOMEBREW_NO_CLEANUP_FORMULAE allows specifying specific formulae to not clean up.

HOMEBREW_NO_INSTALL_UPGRADE
If set, **brew install <formula>** will not upgrade **<formula>** if it is installed but outdated.

HOMEBREW_PRY
If set, use Pry for the **brew irb** command.

HOMEBREW_SIMULATE_MACOS_ON_LINUX
If set, running Homebrew on Linux will simulate certain macOS code paths. This is useful when auditing macOS formulae while on Linux.

HOMEBREW_SSH_CONFIG_PATH
If set, Homebrew will use the given config file instead of **~/.ssh/config** when fetching **git** repos over **ssh**.

Default: **$HOME/.ssh/config**

HOMEBREW_SKIP_OR_LATER_BOTTLES
If set along with **HOMEBREW_DEVELOPER**, do not use bottles from older versions of macOS. This is useful in development on new macOS versions.

HOMEBREW_SORBET_RUNTIME
If set, enable runtime typechecking using Sorbet.

HOMEBREW_SVN
Use this as the **svn**(1) binary.

Default: A Homebrew−built Subversion (if installed), or the system−provided binary.

HOMEBREW_TEMP
Use this path as the temporary directory for building packages. Changing this may be needed if your system temporary directory and Homebrew prefix are on different volumes, as macOS has trouble moving symlinks across volumes when the target does not yet exist. This issue typically occurs when using FileVault or custom SSD configurations.

Default: macOS: **/private/tmp**, Linux: **/tmp**.

HOMEBREW_UPDATE_REPORT_ONLY_INSTALLED
If set, **brew update** only lists updates to installed software.

HOMEBREW_UPDATE_TO_TAG
If set, always use the latest stable tag (even if developer commands have been run).

HOMEBREW_VERBOSE
If set, always assume **−−verbose** when running commands.

HOMEBREW_DEBUG
If set, always assume **−−debug** when running commands.

HOMEBREW_VERBOSE_USING_DOTS
If set, verbose output will print a **.** no more than once a minute. This can be useful to avoid long−running Homebrew commands being killed due to no output.

all_proxy
Use this SOCKS5 proxy for **curl**(1), **git**(1) and **svn**(1) when downloading through Homebrew.

ftp_proxy
Use this FTP proxy for **curl**(1), **git**(1) and **svn**(1) when downloading through Homebrew.

http_proxy
Use this HTTP proxy for **curl**(1), **git**(1) and **svn**(1) when downloading through Homebrew.

https_proxy
Use this HTTPS proxy for **curl**(1), **git**(1) and **svn**(1) when downloading through Homebrew.

no_proxy
A comma−separated list of hostnames and domain names excluded from proxying by **curl**(1), **git**(1) and **svn**(1) when downloading through Homebrew.

SUDO_ASKPASS
If set, pass the **−A** option when calling **sudo**(8).

USING HOMEBREW BEHIND A PROXY

Set the **http_proxy**, **https_proxy**, **all_proxy**, **ftp_proxy** and/or **no_proxy** environment variables documented above.

For example, to use an unauthenticated HTTP or SOCKS5 proxy:

```
export http_proxy=http://$HOST:$PORT

export all_proxy=socks5://$HOST:$PORT
```

And for an authenticated HTTP proxy:

```
export http_proxy=http://$USER:$PASSWORD@$HOST:$PORT
```

SEE ALSO

Homebrew Documentation: *https://docs.brew.sh*

Homebrew API: *https://rubydoc.brew.sh*

git(1), **git−log**(1)

AUTHORS

Homebrew´s Project Leader is Mike McQuaid.

Homebrew´s Project Leadership Committee is Issy Long, Jonathan Chang, Markus Reiter, Misty De Meo and Sean Molenaar.

Homebrew´s Technical Steering Committee is Bo Anderson, FX Coudert, Michka Popoff, Mike McQuaid and Rylan Polster.

Homebrew´s other current maintainers are Alexander Bayandin, Bevan Kay, Branch Vincent, Caleb Xu, Carlo Cabrera, Connor, Daniel Nachun, Dawid Dziurla, Dustin Rodrigues, Eric Knibbe, George Adams, Maxim Belkin, Miccal Matthews, Michael Cho, Nanda H Krishna, Randall, Sam Ford, Shaun Jackman, Steve Peters, Thierry Moisan, Tom Schoonjans, Vitor Galvao and rui.

Former maintainers with significant contributions include Claudia Pellegrino, Seeker, William Woodruff, Jan Viljanen, JCount, commitay, Dominyk Tiller, Tim Smith, Baptiste Fontaine, Xu Cheng, Martin Afanasjew, Brett Koonce, Charlie Sharpsteen, Jack Nagel, Adam Vandenberg, Andrew Janke, Alex Dunn, neutric, Tomasz Pajor, Uladzislau Shablinski, Alyssa Ross, ilovezfs, Chongyu Zhu and Homebrew´s creator: Max Howell.

BUGS

See our issues on GitHub:

Homebrew/brew
https://github.com/Homebrew/brew/issues

Homebrew/homebrew−core
https://github.com/Homebrew/homebrew−core/issues

Homebrew/homebrew−cask
https://github.com/Homebrew/homebrew−cask/issues