



Installation and Setup

Installation

To install Homebrew, open your terminal and run the following command:

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebr
ew/install/HEAD/install.sh)"
```

This script will guide you through the installation process. Follow the prompts, and enter your password if required.

Basic Setup

After installation, it's important to add Homebrew to your `PATH` so that you can run brew commands from any terminal location.

Add these lines to your `.zshrc` or `.bashrc` file:

```
export PATH="/opt/homebrew/bin:$PATH"
```

Then, apply the changes to your current session:

```
source ~/.zshrc # or source ~/.bashrc
```

Alternatively, for interactive setup in `zsh`, run:

```
eval "$(/opt/homebrew/bin/brew
shellenv)"
```

Verify Installation

To ensure Homebrew is installed correctly, run:

```
brew doctor
```

This command will diagnose common issues and provide solutions.

If any problems are reported, follow the suggested steps to resolve them before proceeding.

Package Management

Installing Packages

`brew install` Installs the specified package.

`<package_name>`

`>`

Example:

```
brew install git
```

`brew reinstall` Reinstalls the specified package. Useful for fixing broken installations.

`>`

Example:

```
brew reinstall git
```

`brew install <formula>@<version>` Installs a specific version of a formula.

`>`

Example:

```
brew install node@16
```

Updating Packages

`brew update` Updates Homebrew's package list.

`brew upgrade` Upgrades all outdated packages.

`brew upgrade <package_name>` Upgrades a specific package.

Example:

```
brew upgrade git
```

Removing Packages

`brew uninstall` Uninstalls the specified package.

`<package_name>`

`>`

Example:

```
brew uninstall git
```

`brew cleanup` Removes old versions of packages from the Homebrew cache.

`brew cleanup <package_name>` Removes old versions of specific packages from the Homebrew cache.

`>`

Example:

```
brew cleanup git
```

Brew Cask

Installing Applications

`brew install --cask` Installs the specified application using Brew Cask.

`>`

Example:

```
brew install --cask
firefox
```

`brew reinstall --cask` Reinstalls the specified application using Brew Cask.

`>`

Example:

```
brew reinstall --cask
firefox
```

Listing Installed Applications

To list all applications installed via Brew Cask, use the following command:

```
brew list --cask
```

Uninstalling Applications

`brew uninstall --cask` Uninstalls the specified application using Brew Cask.

`>`

Example:

```
brew uninstall --cask
firefox
```

Advanced Usage

Searching for Packages

<code>brew search</code> <code><keyword></code>	Searches for packages containing the specified keyword. Example: <code>brew search node</code>
<code>brew search</code> <code>/<regex>/</code>	Searches using a regular expression. Example: <code>brew search /^node.*\/</code>

Package Information

<code>brew</code> <code>info</code> <code><package_name></code>	Displays information about a package, including version, dependencies, and caveats. Example: <code>brew info git</code>
<code>brew</code> <code>deps</code> <code><package_name></code>	Lists the dependencies of a package. Example: <code>brew deps node</code>

Linking and Unlinking

<code>brew link</code> <code><package_name></code>	Links a package's files into <code>/usr/local</code> . Example: <code>brew link git</code>
<code>brew unlink</code> <code><package_name></code>	Unlinks a package's files from <code>/usr/local</code> . Example: <code>brew unlink git</code>