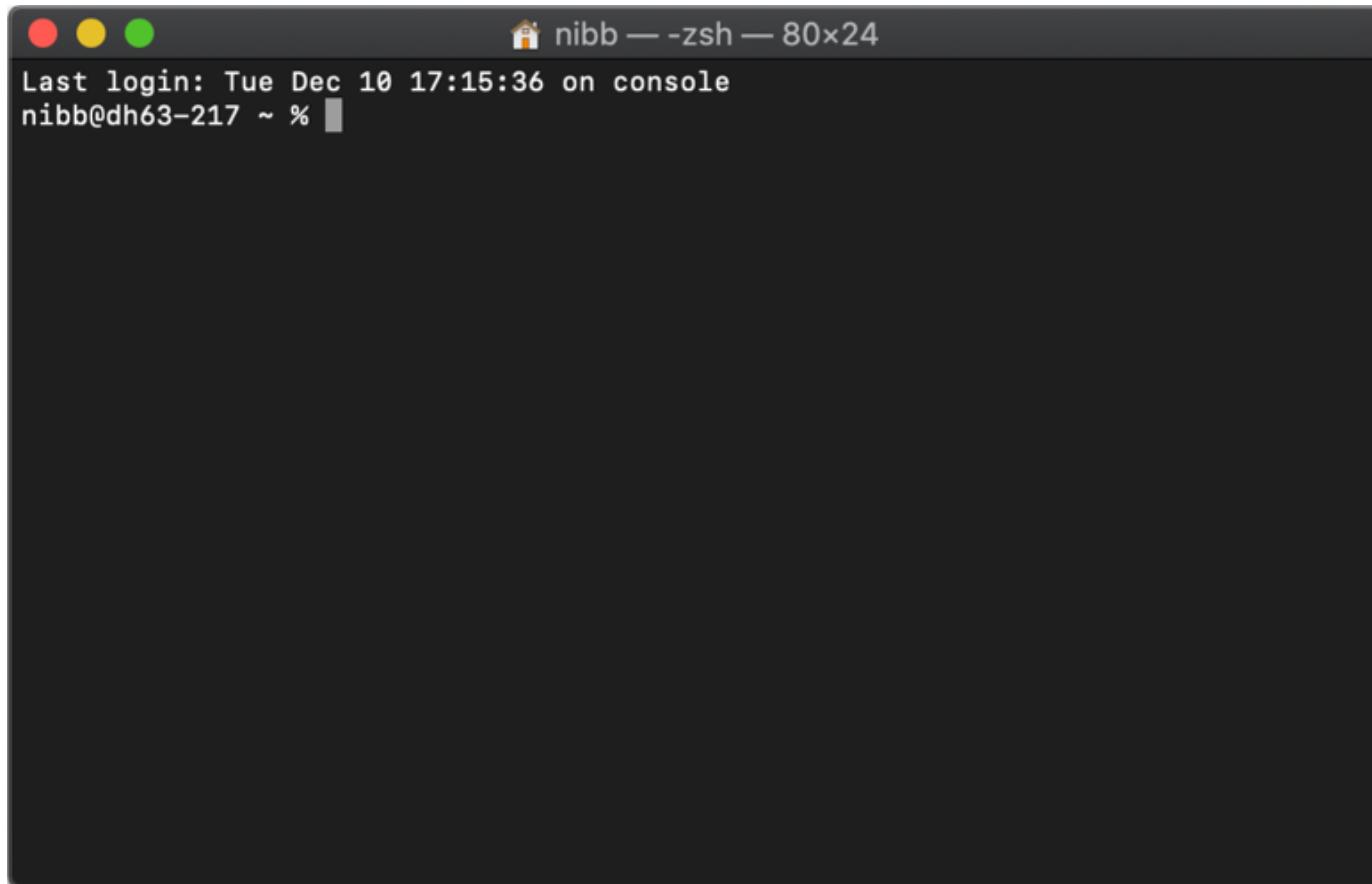


UNIX環境の構築 (macOS編)

Mac環境でUnixコマンドを使用するためには



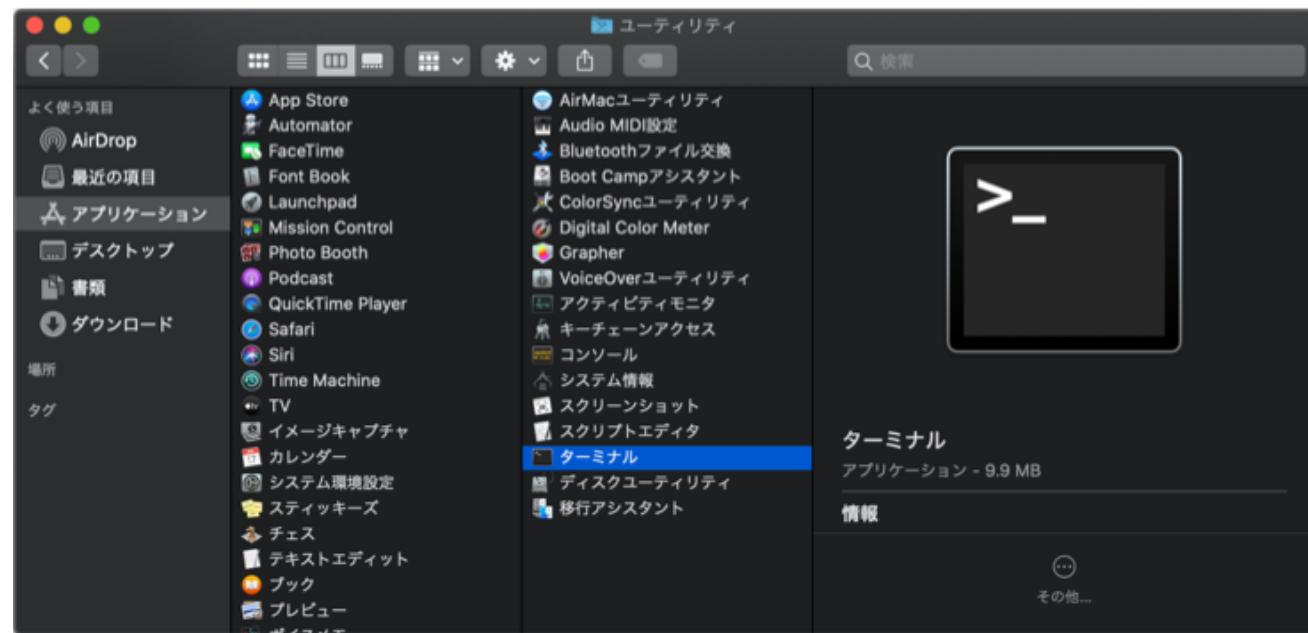
概要

- ▶ macOSは元々Unix系のOSであるDarwin上で動作している。そのため、ターミナル.appから標準的なUnixコマンドを使用することが可能である。
- ▶ しかし、本格的にUnixコマンドを使用した解析操作を行おうとすると、必要なソフトウェアが動作しない・インストールされていないため、つまづくことになる。
- ▶ 本資料はmacOSにおいて必要なソフトウェアをインストールし、解析操作に必要なUnix環境を構築する方法を紹介する。

ターミナル.app



- ▶ [Finder]から[アプリケーション] -> [ユーティリティ]を選択。
- ▶ ターミナル.appを実行することでシェルが起動する。



```
nibb — zsh — 80x24
Last login: Tue Dec 10 17:15:36 on console
nibb@dh63-217 ~ %
```

導入手順の概要

- ▶ 1) Xcode の Command Line Tools をインストールする。
- ▶ 2) パッケージシステムをインストールする。
(Homebrew, Macports, etc)
- ▶ 3) パッケージシステムを用いて、ソフトウェアをインストールする。

Xcode

- ▶ Xcode (<https://developer.apple.com/jp/xcode/>)
- ▶ Xcode はソフトウェアを開発するための統合開発環境である。
- ▶ 米 Apple社から macOS用に無償で配布されている。
- ▶ 今回必要なツールはXcodeの一部機能の Command Line Tools というツールであり、Xcode 全てをインストールする必要はない。
- ▶ (Xcode 全体の容量は7.8GBと巨大である。全てインストールしたい場合はApp Storeから)
- ▶ 今回は Command Line Tools のみをインストールする。

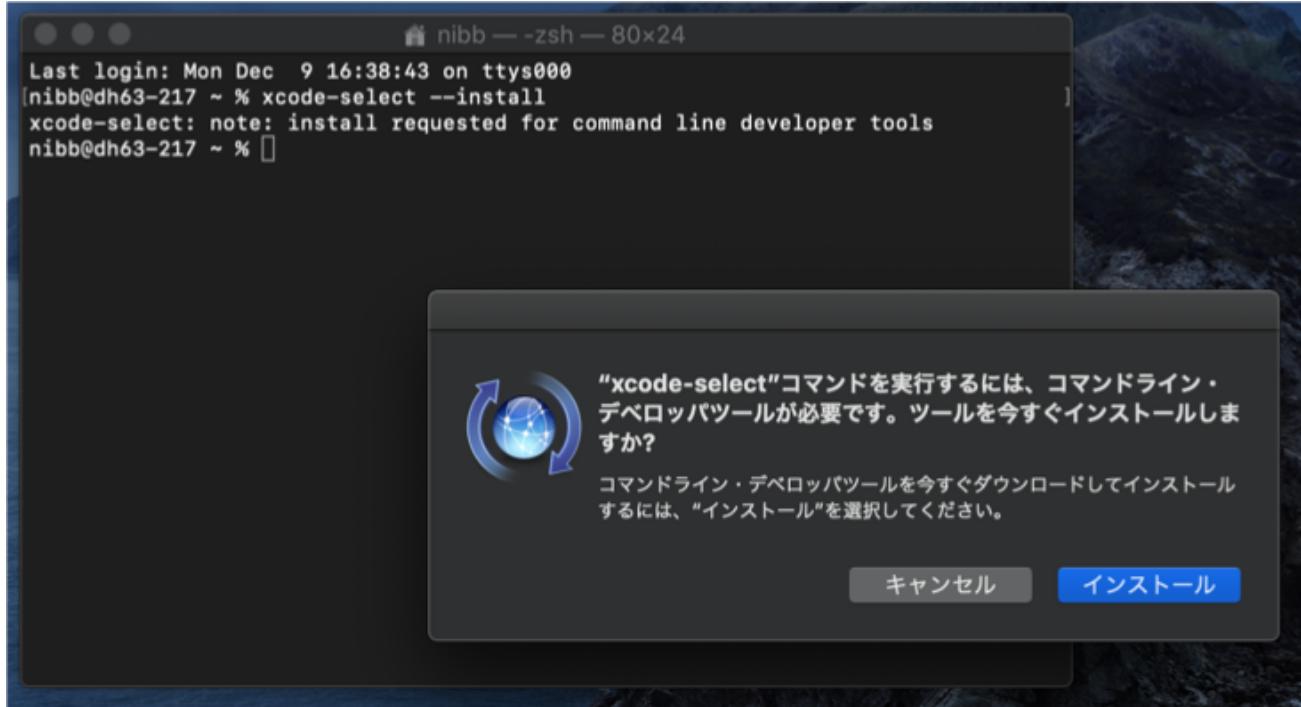


Command Line Tools

- ▶ 前述のXcode は統合開発環境であり、様々なものが同梱されている。
- ▶ その中でソースファイルからコンパイルをするために必要なものは、Xcode に付随する Command Line Tools である。
- ▶ Command Line Tools のみをインストールしたい場合、ターミナル.appから以下のコマンドを入力することでインストールすることができる。

```
xcode-select --install
```

Command Line Tools のインストール



- ▶ xcode の Command Line Tools のみをインストールする場合、ターミナルで以下のコマンドを入力

```
xcode-select --install
```

- ▶ 左図のようなポップアップが現れたら「インストール」を選択する。

パッケージシステム

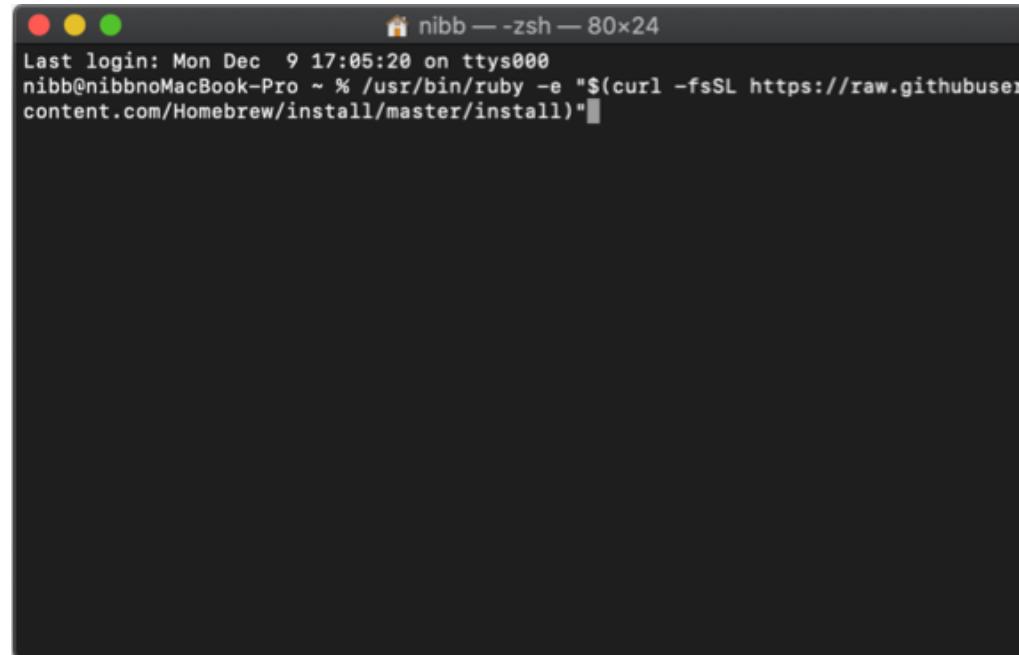
- ▶ Unix環境にソフトウェアをインストールする場合、ソフトウェアに関連づけされているファイル一式をまとめた「パッケージ」を用いることが多い。
- ▶ それらパッケージを用いて、必要なファイルと依存関係も含めてインストールの補助を行ってくれるのが「パッケージシステム」である。
- ▶ macOSで使用できるパッケージシステムは、Homebrew やMacports などが代表例として挙げられる。
- ▶ 本資料ではHomebrew を使用したソフトウェアのインストールを紹介する。

Homebrew



- ▶ Homebrew (https://brew.sh/index_ja)
- ▶ MacOS上でソフトウェアの導入を単純化するパッケージ管理システムの一つ
- ▶ 管理者権限がない一般ユーザでも使用可能
- ▶ パッケージインストール先は /usr/local
- ▶ インストールにかかる時間が比較的少ない

Homebrew のインストール



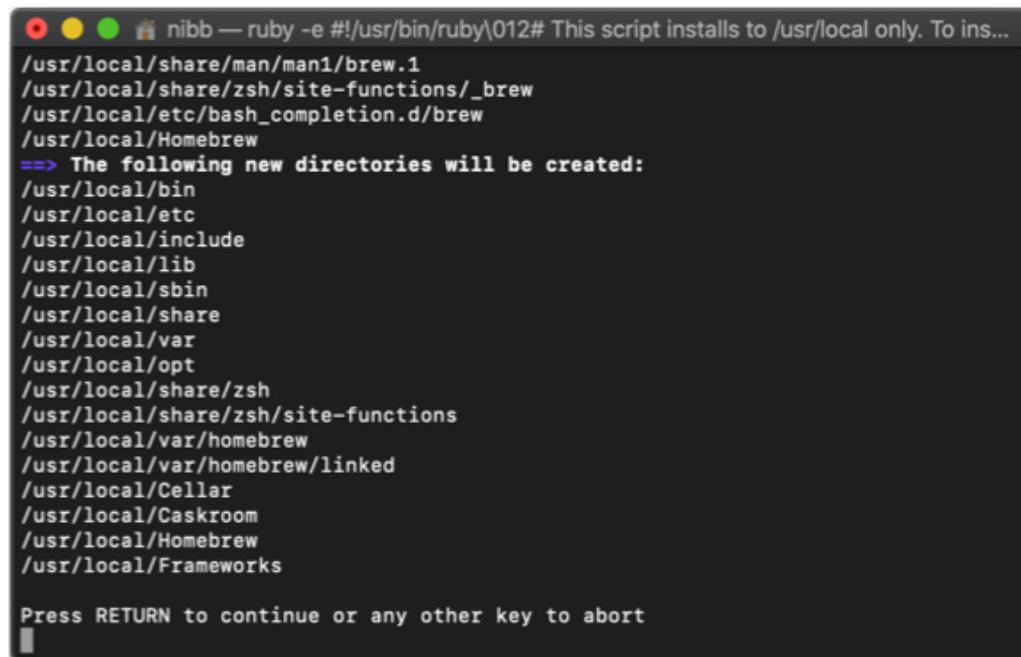
```
nibb — zsh — 80x24
Last login: Mon Dec  9 17:05:20 on ttys000
nibb@nibbnoMacBook-Pro ~ % /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

- ▶ ターミナル.appを開き、以下のコマンドを入力する。

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

(打ち込むコマンドが長いため、https://brew.sh/index_ja からコピー & ペーストするとよい)

Homebrew のインストール



A screenshot of a macOS terminal window titled "nibb — ruby -e #!/usr/bin/ruby\012# This script installs to /usr/local only. To ins...". The window shows the following text:

```
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
==> The following new directories will be created:
/usr/local/bin
/usr/local/etc
/usr/local/include
/usr/local/lib
/usr/local/sbin
/usr/local/share
/usr/local/var
/usr/local/opt
/usr/local/share/zsh
/usr/local/share/zsh/site-functions
/usr/local/var/homebrew
/usr/local/var/homebrew/linked
/usr/local/Cellar
/usr/local/Caskroom
/usr/local/Homebrew
/usr/local/Frameworks

Press RETURN to continue or any other key to abort
```

The terminal window has the standard macOS title bar with red, yellow, and green buttons.

- ▶ リターンキーを押してHomebrewのインストールを続行する。

Homebrew のインストール

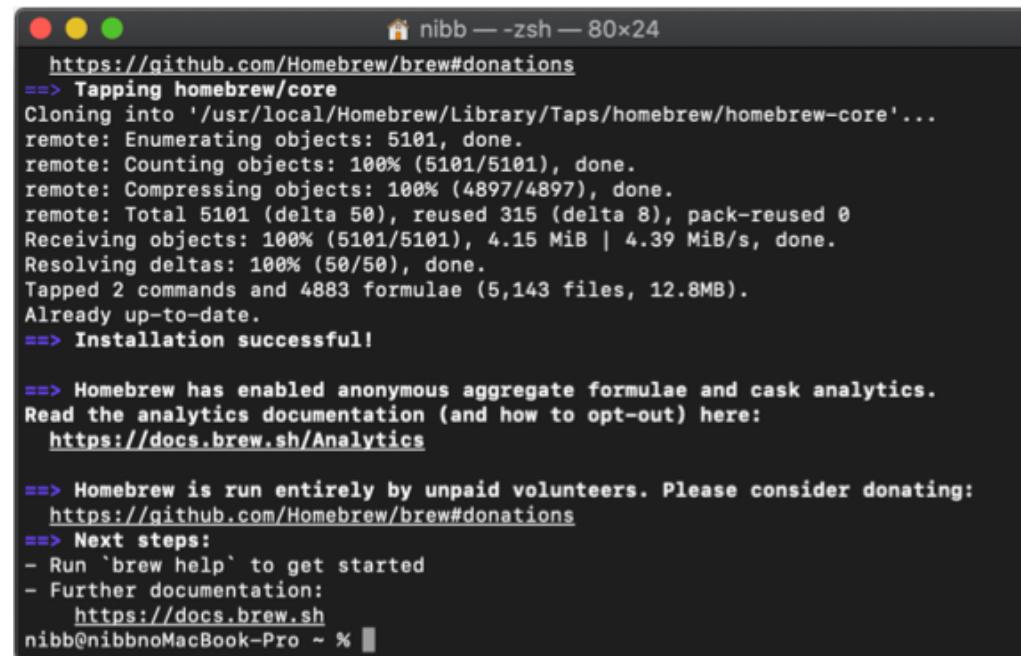
```
● ○ ● nibb — sudo -e ruby -e #!/usr/bin/ruby\012# This script installs to /usr/local only...
/usr/local/bin
/usr/local/etc
/usr/local/include
/usr/local/lib
/usr/local/sbin
/usr/local/share
/usr/local/var
/usr/local/opt
/usr/local/share/zsh
/usr/local/share/zsh/site-functions
/usr/local/var/homebrew
/usr/local/var/homebrew/linked
/usr/local/Cellar
/usr/local/Caskroom
/usr/local/Homebrew
/usr/local/Frameworks

Press RETURN to continue or any other key to abort
==> /usr/bin/sudo /bin/mkdir -p /usr/local/bin /usr/local/etc /usr/local/include
/usr/local/lib /usr/local/sbin /usr/local/share /usr/local/var /usr/local/opt /
/usr/local/share/zsh /usr/local/share/zsh/site-functions /usr/local/var/homebrew
/usr/local/var/homebrew/linked /usr/local/Cellar /usr/local/Caskroom /usr/local/
Homebrew /usr/local/Frameworks
Password:
```

- ▶ パスワードを入力する際、
入力した文字列は表示されないが
ちゃんと入力されている。
- ▶ タイプミスしないよう慎重に入力を行う。

- ▶ macOS上で管理者権限を持っている場合、パスワードを入力する。
- ▶ 入力しないとHomebrew自体のインストールができない。

Homebrew のインストール



```
nibb — zsh — 80x24
https://github.com/Homebrew/brew#donations
==> Tapping homebrew/core
Cloning into '/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core'...
remote: Enumerating objects: 5101, done.
remote: Counting objects: 100% (5101/5101), done.
remote: Compressing objects: 100% (4897/4897), done.
remote: Total 5101 (delta 50), reused 315 (delta 8), pack-reused 0
Receiving objects: 100% (5101/5101), 4.15 MiB | 4.39 MiB/s, done.
Resolving deltas: 100% (50/50), done.
Tapped 2 commands and 4883 formulae (5,143 files, 12.8MB).
Already up-to-date.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations
==> Next steps:
- Run `brew help` to get started
- Further documentation:
  https://docs.brew.sh
nibb@nibbnoMacBook-Pro ~ %
```

- ▶ Homebrewのインストールが完了しました。
- ▶ メッセージに従い、`brew help` と入力すると、Homebrewのヘルプが表示されます。

Homebrew の基本操作

```
$ brew search パッケージ名
```

示されるパッケージを検索します。

```
$ brew install パッケージ名
```

示されるパッケージをインストールします。

```
$ brew uninstall パッケージ名
```

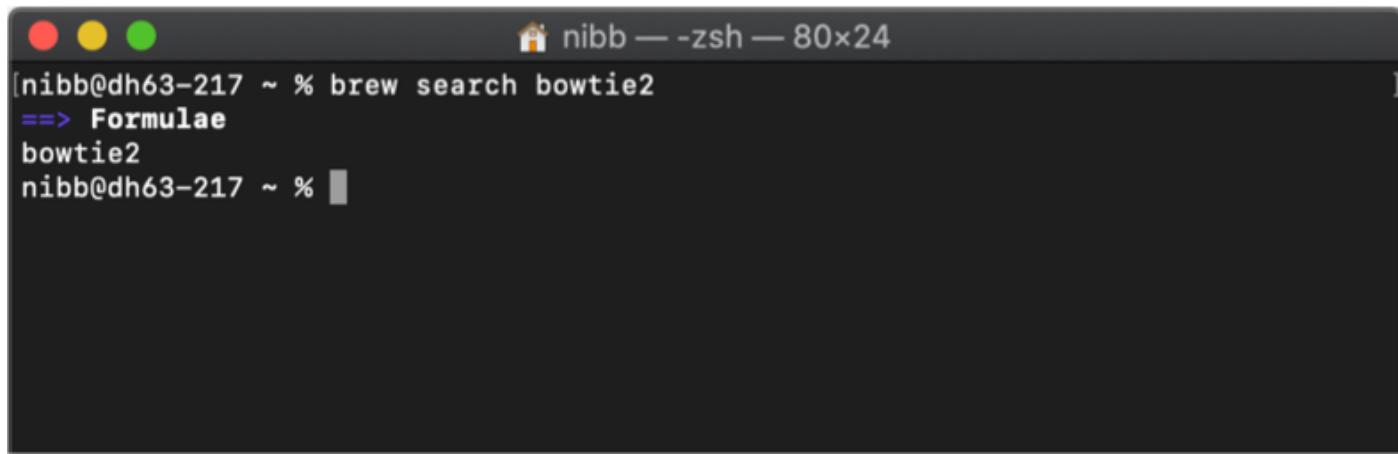
示されるパッケージをアンインストールします。

```
$ brew list
```

インストールされたパッケージ一覧を表示します。

Homebrew の使用例

- ▶ Homebrewを使ってbowtie2をインストールする例を紹介します。
(参照：<http://bowtie-bio.sourceforge.net/bowtie2/index.shtml>)
- ▶ `brew search bowtie2` でbowtie2を探します。



A screenshot of a macOS terminal window titled "nibb — -zsh — 80x24". The window shows the command `brew search bowtie2` being run, followed by the output: "==> Formulae" and "bowtie2". The terminal has a dark background with red, yellow, and green window control buttons.

```
nibb — -zsh — 80x24
[nibb@dh63-217 ~ % brew search bowtie2
==> Formulae
bowtie2
nibb@dh63-217 ~ % ]
```

- ▶ bowtie2というパッケージが配布されていることを確認できました。

Homebrew の使用例

- ▶ bowtie2を実際にインストールします。

```
nibb — zsh — 80x24
[nibb@dh63-217 ~ % brew search bowtie2
=> Formulae
bowtie2
nibb@dh63-217 ~ % brew install bowtie2]
```

brew install bowtie2

```
nibb — zsh — 80x24
echo 'export PATH="/usr/local/opt/sqlite/bin:$PATH"' >> ~/.zshrc

For compilers to find sqlite you may need to set:
export LDFLAGS="-L/usr/local/opt/sqlite/lib"
export CPPFLAGS="-I/usr/local/opt/sqlite/include"

=> python
Python has been installed as
/usr/local/bin/python3

Unversioned symlinks `python', `python-config', `pip' etc. pointing to
`python3', `python3-config', `pip3' etc., respectively, have been installed into
/usr/local/opt/python/libexec/bin

If you need Homebrew's Python 2.7 run
brew install python@2

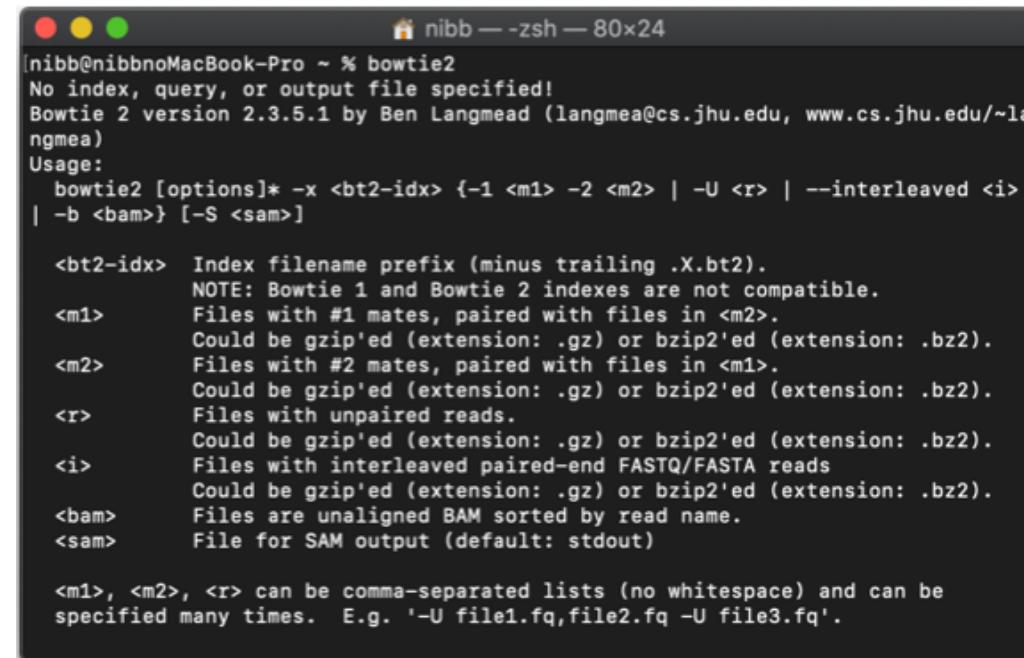
You can install Python packages with
pip3 install <package>
They will install into the site-package directory
/usr/local/lib/python3.7/site-packages

See: https://docs.brew.sh/Homebrew-and-Python
nibb@nibbnoMacBook-Pro ~ %
```

- ▶ インストールできたようです。
- ▶ 実際にbowtie2を実行してみましょう。

bowtie2

Homebrew の使用例



```
nibb — zsh — 80x24
[nibb@nibbnoMacBook-Pro ~ % bowtie2
No index, query, or output file specified!
Bowtie 2 version 2.3.5.1 by Ben Langmead (langmea@cs.jhu.edu, www.cs.jhu.edu/~langmea)
Usage:
  bowtie2 [options]* -x <bt2-idx> { -1 <m1> -2 <m2> | -U <r> | --interleaved <i>
  | -b <bam>} [-S <sam>]

  <bt2-idx>  Index filename prefix (minus trailing .X.bt2).
              NOTE: Bowtie 1 and Bowtie 2 indexes are not compatible.
  <m1>      Files with #1 mates, paired with files in <m2>.
              Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <m2>      Files with #2 mates, paired with files in <m1>.
              Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <r>       Files with unpaired reads.
              Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <i>       Files with interleaved paired-end FASTQ/FASTA reads
              Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <bam>    Files are unaligned BAM sorted by read name.
  <sam>    File for SAM output (default: stdout)

  <m1>, <m2>, <r> can be comma-separated lists (no whitespace) and can be
  specified many times. E.g. '-U file1.fq,file2.fq -U file3.fq'.
```

- ▶ オプションなしで実行したため、bowtie2についての説明文が出てきました。
- ▶ バージョンは2.3.5.1となっています。
- ▶ これでbowtie2をインストールできたことを確認できました。