

2024.3.28版

Pythonのインストール手順を示します。  
スライドはMac環境 (Sonoma 14.4.1)  
でのスクリーンショットです。ウェブ  
ブラウザによって挙動が多少異なりま  
すのでご注意ください。Chrome(推奨)  
で動作確認しています。

# Pythonのインストール (Mac版)

東京大学大学院農学生命科学研究科

アグリバイオインフォマティクス教育研究プログラム

<https://www.iu.a.u-tokyo.ac.jp/>

メール: [info@iu.a.u-tokyo.ac.jp](mailto:info@iu.a.u-tokyo.ac.jp)

X(旧Twitter): @Agribio\_utokyo

# Contents

- Pythonのバージョン確認
- Python本体のインストール
- インストール確認
- Pythonの起動と終了(optional)
- 基本的な利用法(optional)
- パッケージのインストール(optional)
- Biopythonの基本的な利用法(optional)

# Pythonのバージョン確認

最新版Pythonをインストールする前に、すでにインストールされているPythonの有無とそのバージョンを確認しておきます。ターミナルを起動し、①pythonと入力。②pythonに紐づけられたコマンドは無いことが示されました。

```
student --zsh -- 88x28
Last login: Thu Mar 28 10:49:30 on ttys000
~ % python
zsh: command not found: python
~ %
```

# Pythonのバージョン確認

③python3と入力。④developerツールが無いのでインストールが必要だというメッセージとともに、⑤「今すぐインストールしますか？」というメッセージが表示されたら、⑥インストールをクリック。

```
student --zsh -- 88x28
Last login: Thu Mar 28 10:49:30 on ttys000
~ % python
zsh: python: command not found: python
~ % python3
xcode-select: note: No developer tools were found, requesting install.
If developer tools are located at a non-default location on disk, use `sudo xcode-select --switch path/to/Xcode.app` to specify the Xcode that you wish to use for command line developer tools, and cancel the installation dialog.
See `man xcode-select` for more details.
~ %
```

**“python3” コマンドを実行するには、コマンドラインデベロッパツールが必要です。ツールを今すぐインストールしますか？**

コマンドラインデベロッパツールを今すぐダウンロードしてインストールするには、“インストール” を選択してください。

キャンセル    インストール

# Pythonのバージョン確認

③python3と入力。④developerツールが無いのでインストールが必要だというメッセージとともに、⑤「今すぐインストールしますか？」というメッセージが表示されたら、⑥インストールをクリック。使用許諾契約が表示されたら同意するをクリックすると、⑦インストールを開始します。

```
student --zsh -- 88x28
Last login: Thu Mar 28 10:49:30 on ttys000
~ % python
zsh: python: command not found: python
~ % python3
xcode-select: note: No developer tools were found, requesting installation of developer tools.
If developer tools are located at a non-default location on disk, use the
--switch path/to/Xcode.app` to specify the Xcode that you wish to use for command line
developer tools, and cancel the installation dialog.
See `man xcode-select` for more details.
~ %
```

ソフトウェアをインストール中

残り約45秒

中止

# Pythonのバージョン確認

③python3と入力。④developerツールが無いのでインストールが必要だというメッセージとともに、⑤「今すぐインストールしますか？」というメッセージが表示されたら、⑥インストールをクリック。使用許諾契約が表示されたら同意するをクリックすると、⑦インストールを開始します。インストール完了したら⑧完了をクリック。

```
student --zsh -- 88x28
Last login: Thu Mar 28 10:49:30 on ttys000
~ % python
zsh: python: command not found: python
~ % python3
xcode-select: note: No developer tools were found, requesting installation.
If developer tools are located at a non-default location on disk, use the
option `--switch path/to/Xcode.app` to specify the Xcode that you wish to
install. Otherwise, use the command to install developer tools, and cancel the
installation dialog.
See `man xcode-select` for more details.
~ %
```



ソフトウェアがインストールされました。



完了

# Pythonのバージョン確認

③python3と入力。④developerツールが無いのでインストールが必要だというメッセージとともに、⑤「今すぐインストールしますか?」というメッセージが表示されたら、⑥インストールをクリック。使用許諾契約が表示されたら同意するをクリックすると、⑦インストールを開始します。インストール完了したら⑧完了をクリック。もう一度⑨python3と実行してみると、今度は起動しました。(XcodeをインストールしなくてもPythonが起動してきたので、ここではこのままXcodeはインストールせずに進みます。)

student — Python — 88x28

Last login: Thu Mar 28 17:04:56 on ttys000

~ % python

zsh: command not found: python

~ % python3

xcode-select: note: No developer tools were found, requesting installation. If developer tools are located at a non-default location on disk, use the option `--switch path/to/Xcode.app` to specify the Xcode that you wish to use. See `man xcode-select` for more details.

~ % python3

Python 3.9.6 (default, Feb 3 2024, 15:58:28)

[Clang 15.0.0 (clang-1500.3.9.4)] on darwin

Type "help", "copyright", "credits" or "license" for more information

>>> █

# Pythonのバージョン確認

- ①起動したPythonのバージョンは、**3.9.6**であることが確認できました。
- ②`quit()`と入力してPythonを終了。

```
student --zsh -- 88x28
Last login: Thu Mar 28 17:04:56 on ttys000
~ % python
zsh: command not found: python
~ % python3
xcode-select: note: No developer tools were found, requesting install.
If developer tools are located at a non-default location on disk, use `sudo xcode-select
--switch path/to/Xcode.app` to specify the Xcode that you wish to use for command line
developer tools, and cancel the installation dialog.
See `man xcode-select` for more details.
~ % python3
Python 3.9.6 (default, Feb  3 2024, 15:58:28)
[Clang 15.0.0 (clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()
~ %
```



# Pythonのバージョン確認

③which python3と入力し、Python3の実体がどこにあるかを確認し④  
/usr/bin/python3であることが確認できました。なお、このPython3.8.2は、システムやプレインストールされているソフトウェアが使用している可能性があります、削除せずに残しておきます。

```
student --zsh -- 88x28
Last login: Thu Mar 28 17:04:56 on ttys000
~ % python
zsh: command not found: python
~ % python3
xcode-select: note: No developer tools were found, requesting install.
If developer tools are located at a non-default location on disk, use `sudo xcode-select
--switch path/to/Xcode.app` to specify the Xcode that you wish to use for command line
developer tools, and cancel the installation dialog.
See `man xcode-select` for more details.
~ % python3
Python 3.9.6 (default, Feb  3 2024, 15:58:28)
[Clang 15.0.0 (clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> quit()
~ % which python3
/usr/bin/python3
~ %
```

③

④

# Contents

- Pythonのバージョン確認
- Python本体のインストール
- インストール確認
- Pythonの起動と終了(optional)
- 基本的な利用法(optional)
- パッケージのインストール(optional)
- Biopythonの基本的な利用法(optional)

# Pythonのインストール

最新版Pythonを使用するためのインストール方法を説明します。①Pythonのサイト。②Downloads...

Welcome to Python.org

python.org

Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233
377 610 987
```

### Functions Defined

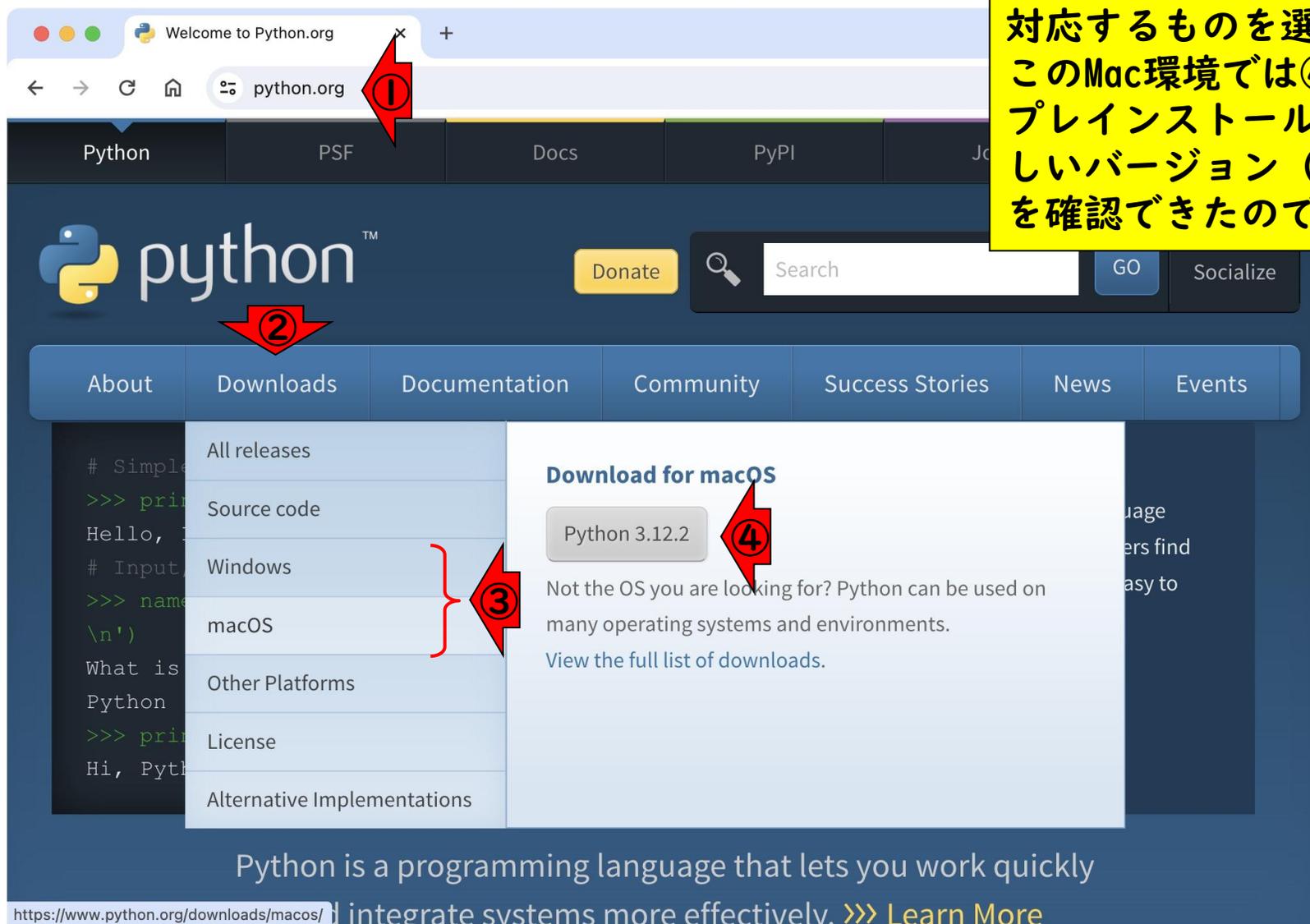
The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

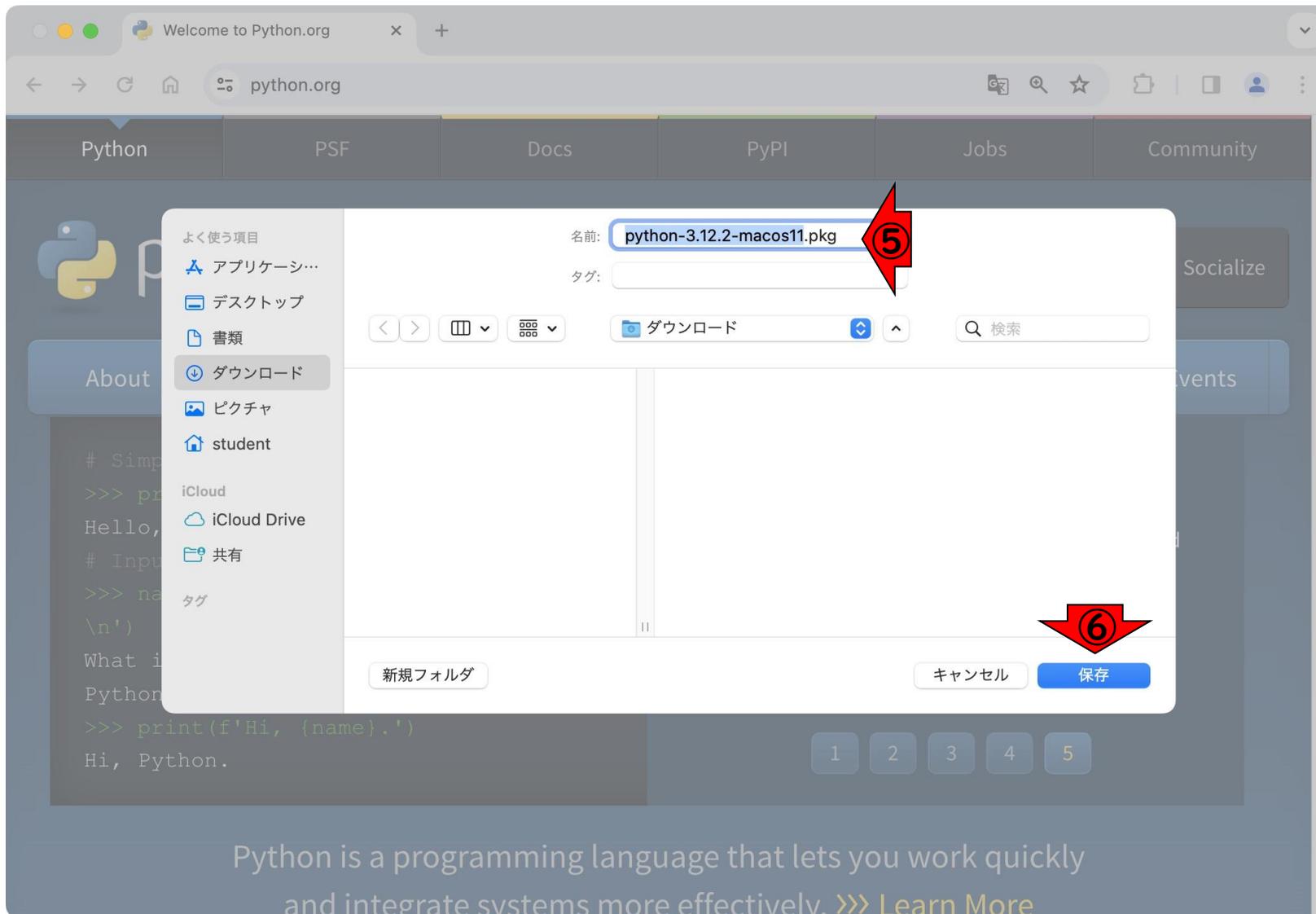
# Pythonのインストール2

最新版Pythonを使用するためのインストール方法を説明します。①Pythonのサイト。②Downloads。③各自のOSに対応するものを選べばよいが、例えばこのMac環境では④のように見えて、プレインストール版(3.9.6)より新しいバージョン(3.12.2)であることを確認できたので、④をクリック。



# Pythonのインストール3

⑤pkgファイルを任意の場所に、⑥保存。



# Pythonのインストール4

ダウンロードしたpkgファイルをダブルクリックした状態。①続けるをクリック、

## Pythonのインストール

ようこそPythonインストーラへ

This package will install **Python 3.12.2** for **macOS 10.9 or later**.

**Python for macOS** consists of the [Python](#) programming language interpreter and its batteries-included standard library to allow easy access to macOS features. It also includes the Python integrated development environment, **IDLE**. You can also use the included **pip** to download and install third-party packages from the [Python Package Index](#).

At the end of this install, click on **Install Certificates** to install a set of current SSL root certificates.

### ● はじめに

- 大切な情報
- 使用許諾契約
- インストール先
- インストールの種類
- インストール
- 概要



戻る

続ける

# Pythonのインストール5

ダウンロードしたexeファイルをダブルクリックした状態。①続けるをクリック、②続ける、

## Pythonのインストール

### 大切な情報

This package will install Python 3.12.2 for macOS 10.9 or later for the following architecture(s): arm64, x86\_64.

#### Certificate verification and OpenSSL

This package includes its own private copy of OpenSSL 3.0. The trust certificates in system and user keychains managed by the *Keychain Access* application and the *security* command line utility are not used as defaults by the Python `ssl` module. A sample command script is included in `/Applications/Python 3.12` to install a curated bundle of default root certificates from the third-party `certifi` package (<https://pypi.org/project/certifi/>). Double-click on `Install Certificates` to run it.

The bundled `pip` has its own default certificate store for verifying download connections.

#### Install Options

You can control some aspects of what is installed by this package. To see the options, click on the `Customize` button in the `Installation Type` step of the macOS installer app. Click on a package name in the list shown to see more information about that

プリント...

保存...

戻る

続ける

● はじめに

● **大切な情報**

● 使用許諾契約

● インストール先

● インストールの種類

● インストール

● 概要

# Pythonのインストール6

ダウンロードしたexeファイルをダブルクリックした状態。①続けるをクリック、②続ける、③続ける。

Pythonのインストール

使用許諾契約

## HISTORY AND LICENSE

### HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <https://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <https://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see <https://www.zope.dev>). In 2001, the Python Software Foundation (PSF, see <https://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

プリント...

保存...

戻る

続ける

- はじめに
- 大切な情報
- 使用許諾契約

- インストール先
- インストールの種類
- インストール
- 概要

# Pythonのインストール7

①使用許諾契約を確認後、②同意するをクリック。

使用許諾契約

## HISTORY AND LICENSE

### HISTORY OF THE SOFTWARE

このソフトウェアのインストールを続けるには、ソフトウェア使用許諾契約の条件に同意する必要があります。

インストールを続けるには、“同意する”をクリックしてください。インストールをキャンセルしてインストーラを終了する場合は、“同意しない”をクリックしてください。



使用許諾契約を読む



同意しない

同意する

Zope Corporation, see <https://www.zope.dev>). In 2001, the Python Software Foundation (PSF, see <https://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

プリント...

保存...

戻る

続ける

# Pythonのインストール8

Pythonのインストール

“Macintosh HD” に標準インストール

この操作には、コンピュータ上に178 MBの領域が必要です。

ディスク“Macintosh HD”にこのソフトウェアを標準インストールするには、“インストール”をクリックしてください。

- はじめに
- 大切な情報
- 使用許諾契約
- インストール先
- **インストールの種類**
- インストール
- 概要



カスタマイズ

戻る

インストール

# Pythonのインストール9

①パスワードを入力、②ソフトウェアをインストールをクリック。



# Pythonのインストール…10

①パスワードを入力、②ソフトウェアをインストールをクリック。インストール中…

Pythonのインストール

Pythonのインストール

- はじめに
- 大切な情報
- 使用許諾契約
- インストール先
- インストールの種類
- **インストール**
- 概要

パッケージスクリプトを実行中…



戻る

続ける

# Pythonのインストール…II

①パスワードを入力、②ソフトウェアをインストールをクリック。インストール中。③無事完了したら、④閉じる。

Pythonのインストール

インストールが完了しました。

③

Congratulations! **Python 3.12.2 for macOS 10.9 or later** was successfully installed.

One more thing: to verify the identity of secure network connections, this Python needs a set of SSL root certificates. You can download and install a current curated set from [the Certifi project](#) by double-clicking on the Install Certificates icon in [the Finder window](#). See [the ReadMe file](#) for more information.

- はじめに
- 大切な情報
- 使用許諾契約
- インストール先
- インストールの種類
- インストール
- 概要

④

戻る

閉じる

# Pythonのインストール…12

①パスワードを入力、②ソフトウェアをインストールをクリック。インストール中。③無事完了したら、④閉じる。インストーラは⑤ゴミ箱に入れるを選んで構いません。

Pythonのインストール

インストールが完了しました。

Congratulations! Python 3.12.2 for macOS 10.9 or later was successfully installed.

- はじめに
- 大切な情報
- 使用許諾契約
- インストール先
- インストールの種類
- インストール
- 概要



“Python”のインストーラをゴミ箱に入れますか？

このパッケージを現在の場所に残す場合は、“残す”をクリックしてください。

⑤

残す

ゴミ箱に入れる

戻る

閉じる

# Pythonのインストール…13

インストール終了と同時に、Python3.12が①アプリケーションフォルダの下にインストールされたことが示されました。

The screenshot shows a Finder window titled "Python 3.12" with the following contents:

名前	変更日	サイズ	種類
Icon?	2024年2月7日 7:12	567 KB	書類
IDLE	今日 18:13	188 KB	アプリケーション
Install Certificates.command	2024年2月7日 7:12	1 KB	ターミナ…クリプト
License.rtf	2024年2月7日 7:12	15 KB	リッチテキスト書類
Python Documentation.html	今日 18:13	99 バイト	エイリアス
Python Launcher	今日 18:13	402 KB	アプリケーション
ReadMe.rtf	2024年2月7日 7:12	3 KB	リッチテキスト書類
Update Shell Profile.command	2024年2月7日 7:12	3 KB	ターミナ…クリプト

A red arrow points to the "Python Launcher" application. The breadcrumb path at the bottom reads: Macintosh HD > アプリケーション > Python 3.12.

# Contents

- Pythonのバージョン確認
- Python本体のインストール
- インストール確認
- Pythonの起動と終了(optional)
- 基本的な利用法(optional)
- パッケージのインストール(optional)
- Biopythonの基本的な利用法(optional)

# インストール確認I

```
student - zsh - 88x28
Last login: Thu Mar 28 17:05:41 on ttys000
~ % python3 --version
Python 3.12.2
~ % python3 -V
Python 3.12.2
~ %
```

起動中のターミナルがあれば、終了し、新規にターミナルを起動します。① `python3 --version`と入力し、Pythonのバージョンを確認します。なお、② `python3 -V`でも同様にバージョンを確認することができます。これにより、ダウンロードしてインストールしたPython **3.12.2**が正しくインストールされたことを確認できました。

# インストール確認2

③ 「python3 -h」と打つと、このように見えます。ちなみにインストールされた3.12.2の実体は、④ /Library/Frameworks/Python.framework/Versions/3.12/Resources/Python.app/Contents/MacOS/Pythonであることがわかります。

```
student --zsh -- 88x28
Last login: Thu Mar 28 17:05:41 on ttys000
~ % python3 --version
Python 3.12.2
~ % python3 -h
Python 3.12.2
~ % python3 -h
usage: /Library/Frameworks/Python.framework/Versions/3.12/Resources/Python.app/Contents/MacOS/Python [option] ... [-c cmd | -m mod | file | -] [arg] ...
Options (and corresponding environment variables):
-b      : issue warnings about str(bytes_instance), str(bytearray_instance)
         and comparing bytes/bytearray with str. (-bb: issue errors)
-B      : don't write .pyc files on import; also PYTHONDONTWRITEBYTECODE=x
-c cmd  : program passed in as string (terminates option list)
-d      : turn on parser debugging output (for experts only, only works on
         debug builds); also PYTHONDEBUG=x
-E      : ignore PYTHON* environment variables (such as PYTHONPATH)
-h      : print this help message and exit (also -? or --help)
-i      : inspect interactively after running script; forces a prompt even
         if stdin does not appear to be a terminal; also PYTHONINSPECT=x
-I      : isolate Python from the user's environment (implies -E and -s)
-m mod  : run library module as a script (terminates option list)
-O      : remove assert and __debug__-dependent statements; add .opt-1 before
         .pyc extension; also PYTHONOPTIMIZE=x
-O0     : do -O changes and also discard docstrings; add .opt-2 before
         .pyc extension
-P      : don't prepend a potentially unsafe path to sys.path; also PYTHONSAFEPATH
-q      : don't print version and copyright messages on interactive startup
-s      : don't add user site directory to sys.path; also PYTHONNOUSERSITE
```



# インストール確認3

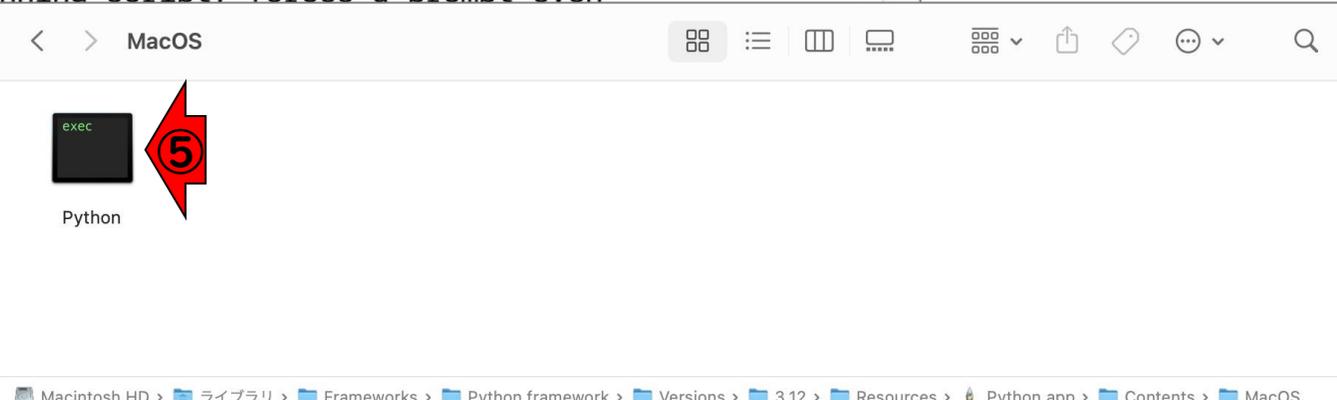
③ 「python3 -h」と打つと、このように見えます。ちなみにインストールされた3.12.2の実体は、④

/Library/Frameworks/Python.framework/Versions/3.12/Resources/Python.app/Contents/MacOS/Pythonであることがわかります。インストールしたPythonの実体⑤をダブルクリックしてみると、

```
student --zsh -- 88x28
Last login: Thu Mar 28 17:05:41 on ttys000
~ % python3 --version
Python 3.12.2
~ % python3 -h
Python 3.12.2
~ % python3 -h

usage: /Library/Frameworks/Python.framework/Versions/3.12/Resources/Python.app/Contents/MacOS/Python [option] ... [-c cmd | -m mod | file | -] [arg] ...

Options (and corresponding environment variables):
-b      : issue warnings about str(bytes_instance), str(bytearray_instance)
         and comparing bytes/bytearray with str. (-bb: issue errors)
-B      : don't write .pyc files on import; also PYTHONDONTWRITEBYTECODE=x
-c cmd  : program passed in as string (terminates option list)
-d      : turn on parser debugging output (for experts only, only works on
         debug builds); also PYTHONDEBUG=x
-E      : ignore PYTHON* environment variables (such as PYTHONPATH)
-h      : print this help message and exit (also -? or --help)
-i      : inspect interactively after running script: forces a prompt even
         if stdin does not terminate the program
-I      : isolate Python from the user's PYTHONPATH, and hide some warnings
-m mod  : run library module as script (terminates option list)
-O      : remove assert statements; compile as if 'OPTIMIZE' flag was set during
         compilation
-O0     : do -O changes and add asserts back
-O1    : do -O changes and add asserts back
-O2    : do -O changes and add asserts back
-P      : don't prepend user site to sys.path
-q      : don't print version number
-s      : don't add user site directory to sys.path
```



# インストール確認4

student — Python — Python — 88x28

⑥

```
Last login: Thu Mar 28 18:20:18 on ttys001
~ % /Library/Frameworks/Python.framework/Versions/3.12/Resources/Py
OS/Python ; exit;
Python 3.12.2 (v3.12.2:6abddd9f6a, Feb  6 2024, 17:02:06) [Clang 13
.30] on darwin
Type "help", "copyright", "credits" or "license" for more informati
>>> █
```

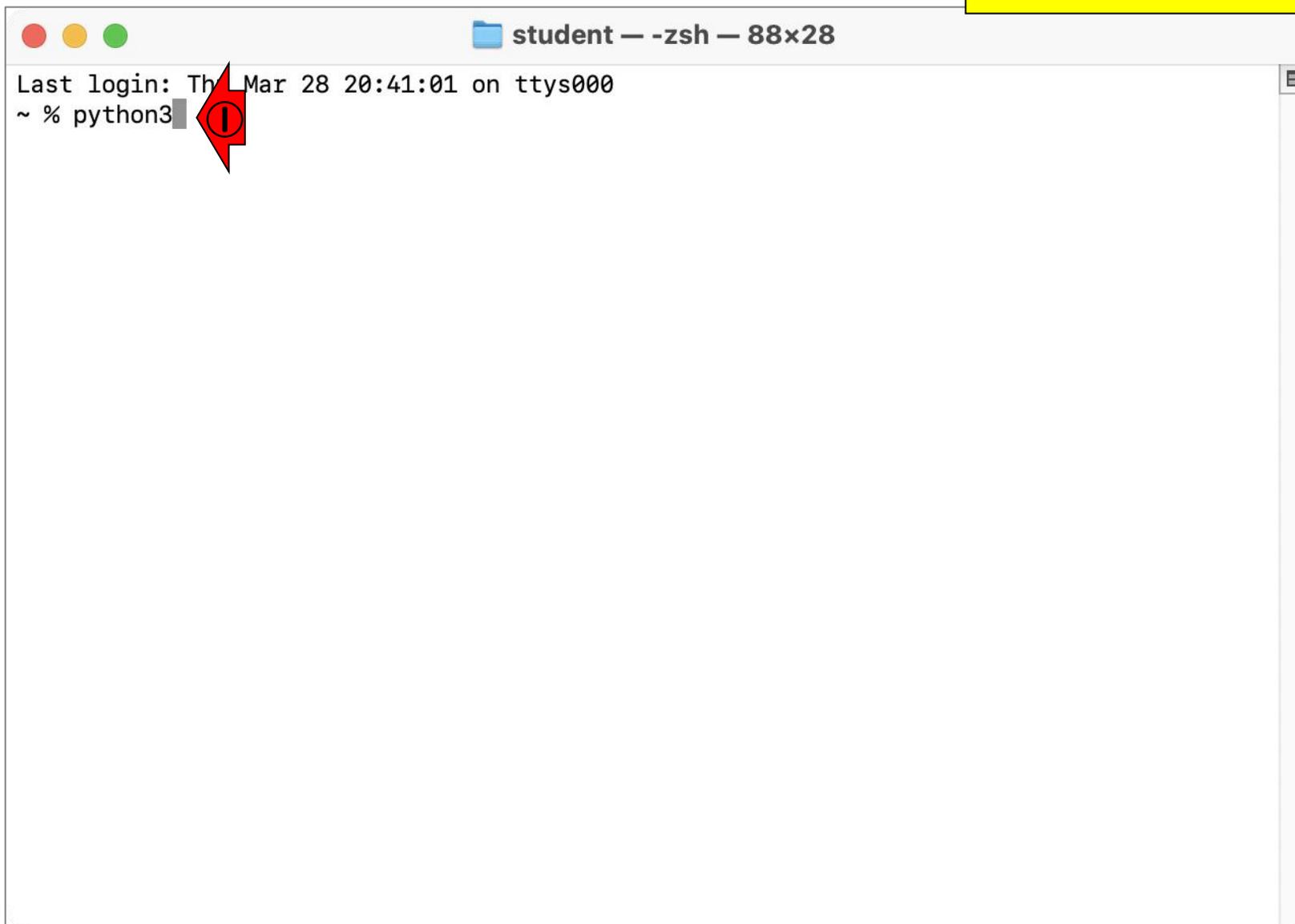
③ 「python3 -h」と打つと、このように見えます。ちなみにインストールされた3.12.2の実体は、④  
/Library/Frameworks/Python.framework/Versions/3.12/Resources/Python.app/Contents/MacOS/Pythonであることがわかります。インストールしたPythonの実体⑤をダブルクリックしてみると、⑥Pythonが起動した状態のターミナルが起動します。このようにインストールしたバージョンのPythonを起動できない場合は、インストールを失敗していると考えられます。

# Contents

- Pythonのバージョン確認
- Python本体のインストール
- インストール確認
- Pythonの起動と終了(optional)
- 基本的な利用法(optional)
- パッケージのインストール(optional)
- Biopythonの基本的な利用法(optional)

# Pythonの起動と終了！

ターミナルを利用してPythonを起動します。起動は、①「python3」と打ってリターン。



A terminal window titled "student - -zsh - 88x28" is shown. The window contains the text "Last login: Thu Mar 28 20:41:01 on ttys00" and the command prompt "~ % python3". A red arrow with a circled '1' points to the command prompt.

```
student - -zsh - 88x28
Last login: Thu Mar 28 20:41:01 on ttys00
~ % python3
```

# Pythonの起動と終了2

こんな感じになればOK。Pythonの場合は、②「>>>」のように「大なりマーク(>)」が3つ連続で見えていれば、Pythonのコマンドを実行できます。

student — Python — 88x28

Last login: Thu Mar 28 20:41:01 on ttys000

~ % python3

Python 3.12.2 (v3.12.2:6abddd9f6a, Feb 6 2024, 17:02:06) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

>>>



# Pythonの起動と終了3

こんな感じになればOK。Pythonの場合は、②「>>>」のように「大なりマーク(>)」が3つ連続で見えていれば、Pythonのコマンドを実行できます。Pythonの終了は、③「quit()」と打ちこんでリターン。

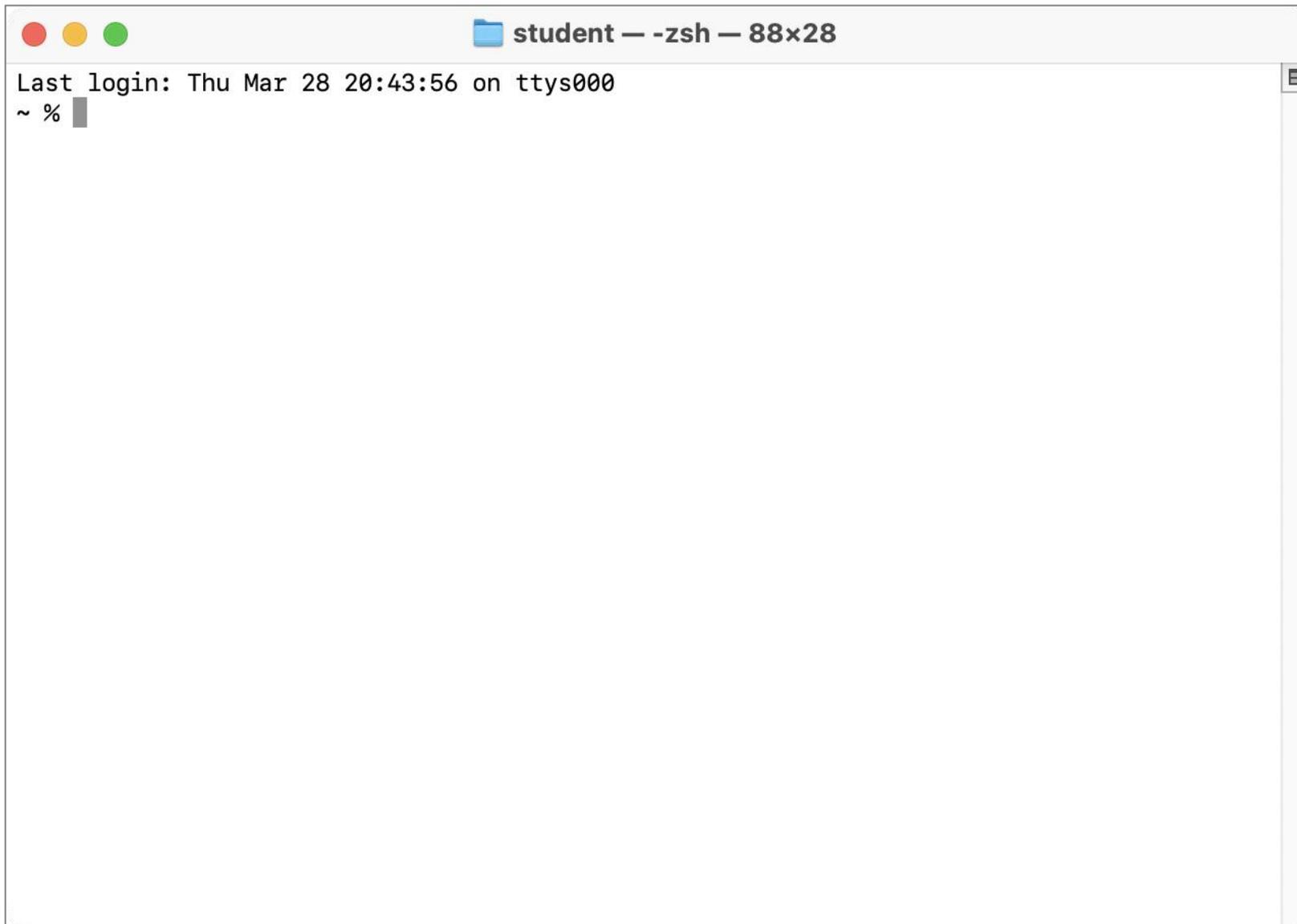
```
student — -zsh — 88x28
Last login: Thu Mar 28 20:41:01 on ttys000
~ % python3
Python 3.12.2 (v3.12.2:6abddd9f6a, Feb  6 2024, 17:02:06) [Clang 13
.30] on darwin
Type "help" "copyright", "credits" or "license" for more information.
[>>> quit()
~ %
```



# Contents

- Pythonのバージョン確認
- Python本体のインストール
- インストール確認
- Pythonの起動と終了(optional)
- **基本的な利用法(optional)**
- パッケージのインストール(optional)
- Biopythonの基本的な利用法(optional)

# 基本的な利用法I

A screenshot of a terminal window. The title bar shows 'student - -zsh - 88x28'. The terminal content displays 'Last login: Thu Mar 28 20:43:56 on ttys000' followed by a prompt '~ %' and a cursor. There is a hamburger menu icon in the top right corner of the terminal area.

```
student - -zsh - 88x28
Last login: Thu Mar 28 20:43:56 on ttys000
~ % █
```

# 基本的な利用法2

①ターミナル起動後の状態。ターミナル上では、②「| + |」を実行しても2という結果が得られないことを確認しているだけです。

```
student --zsh -- 88x28
Last login: Thu Mar 28 20:13:56 on ttys000
~ % 1 + 1
zsh: command not found: 1
~ %
```

# 基本的な利用法3

①ターミナル起動後の状態。ターミナル上では、②「1 + 1」を実行しても2という結果が得られないことを確認しているだけです。③「python3」と打って、Pythonを起動した状態。

student — Python — 88x28

Last login: Thu Mar 28 20:43:56 on ttys000

~ % 1 + 1

zsh: command not found: 1

~ % python3

Python 3.12.2 (v3.12.2:6abddd9f6a, Feb 6 2024, 17:02:06) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

>>> █

# 基本的な利用法4

student — Python — 88x28

Last login: Thu Mar 28 20:43:56 on ttys000

~ % 1 + 1

zsh: command not found: 1

~ % python3

Python 3.12.2 (v3.12.2:6abddd9f6a, Feb 6 2024, 17:02:06) [Clang 13.0.0] on darwin

Type "help" "copyright", "credits" or "license" for more information

[>>> 1 + 1

2

>>> █

④

①ターミナル起動後の状態。ターミナル上では、②「1 + 1」を実行しても2という結果が得られないことを確認しているだけです。③「python3」と打って、Pythonを起動した状態。④「1 + 1」を再度実行すると、正しく2という答えが表示されていることがわかります。同じコマンドプロンプトでも、Python起動の有無がよくわかる例といえます。

# 基本的な利用法5

student — Python — 88x28

Last login: Thu Mar 28 20:43:56 on ttys000

~ % 1 + 1

zsh: command not found: 1

~ % python3

Python 3.12.2 (v3.12.2:6abddd9f6a, Feb 6 2024, 17:02:06) [Clang 13.0.0] on darwin

Type "help", "copyright", "credits" or "license" for more information

>>> 1 + 1

2

>>> aa = 5

>>> aa

5

>>> █



①ターミナル起動後の状態。ターミナル上では、②「1 + 1」を実行しても2という結果が得られないことを確認しているだけです。③「python3」と打って、Pythonを起動した状態。④「1 + 1」を再度実行すると、正しく2という答えが表示されていることがわかります。同じコマンドプロンプトでも、Python起動の有無がよくわかる例といえます。⑤「aa = 5」のように打ち込んで、任意の変数(この場合はaa)に値を代入して利用することもできます。もちろん、次の行で「aa」と打ち込むと、その中身である5が正しく表示されます。

# 基本的な利用法6

```
student --zsh-- 88x28
Last login: Thu Mar 28 20:43:56 on ttys000
~ % 1 + 1
zsh: command not found: 1
~ % python3
Python 3.12.2 (v3.12.2:6abddd9f6a, Feb 6 2024, 17:02:06) [Clang 13
.30] on darwin
Type "help", "copyright", "credits" or "license" for more informati
[>>> 1 + 1
2
[>>> aa = 5
[>>> aa
5
[>>> quit()
~ %
```



①ターミナル起動後の状態。ターミナル上では、②「1 + 1」を実行しても2という結果が得られないことを確認しているだけです。③「python3」と打って、Pythonを起動した状態。④「1 + 1」を再度実行すると、正しく2という答えが表示されていることがわかります。同じコマンドプロンプトでも、Python起動の有無がよくわかる例といえます。⑤「aa = 5」のように打ち込んで、任意の変数(この場合はaa)に値を代入して利用することもできます。もちろん、次の行で「aa」と打ち込むと、その中身である5が正しく表示されます。⑥「quit()」でPythonを終了。

# Contents

- Pythonのバージョン確認
- Python本体のインストール
- インストール確認
- Pythonの起動と終了(optional)
- 基本的な利用法(optional)
- パッケージのインストール(optional)
- Biopythonの基本的な利用法(optional)

# パッケージのインストール

Rとパッケージの関係同様、Pythonも様々なパッケージをインストールして利用します。ここでは、生命科学分野で有名な①Biopythonのインストールを行います。②Downloadをクリック。

Python Tools for Computational Molecular Biology

Documentation  
Download ②  
Mailing lists  
News  
Biopython Contributors  
Scriptcentral  
Source Code  
GitHub project

Biopython version 1.83  
© 2024. All rights

Edit this page on GitHub

## Biopython

See also our [News feed](#).

### Introduction

Biopython is a set of freely available tools for biological computation written in [Python](#) by an international team of developers.

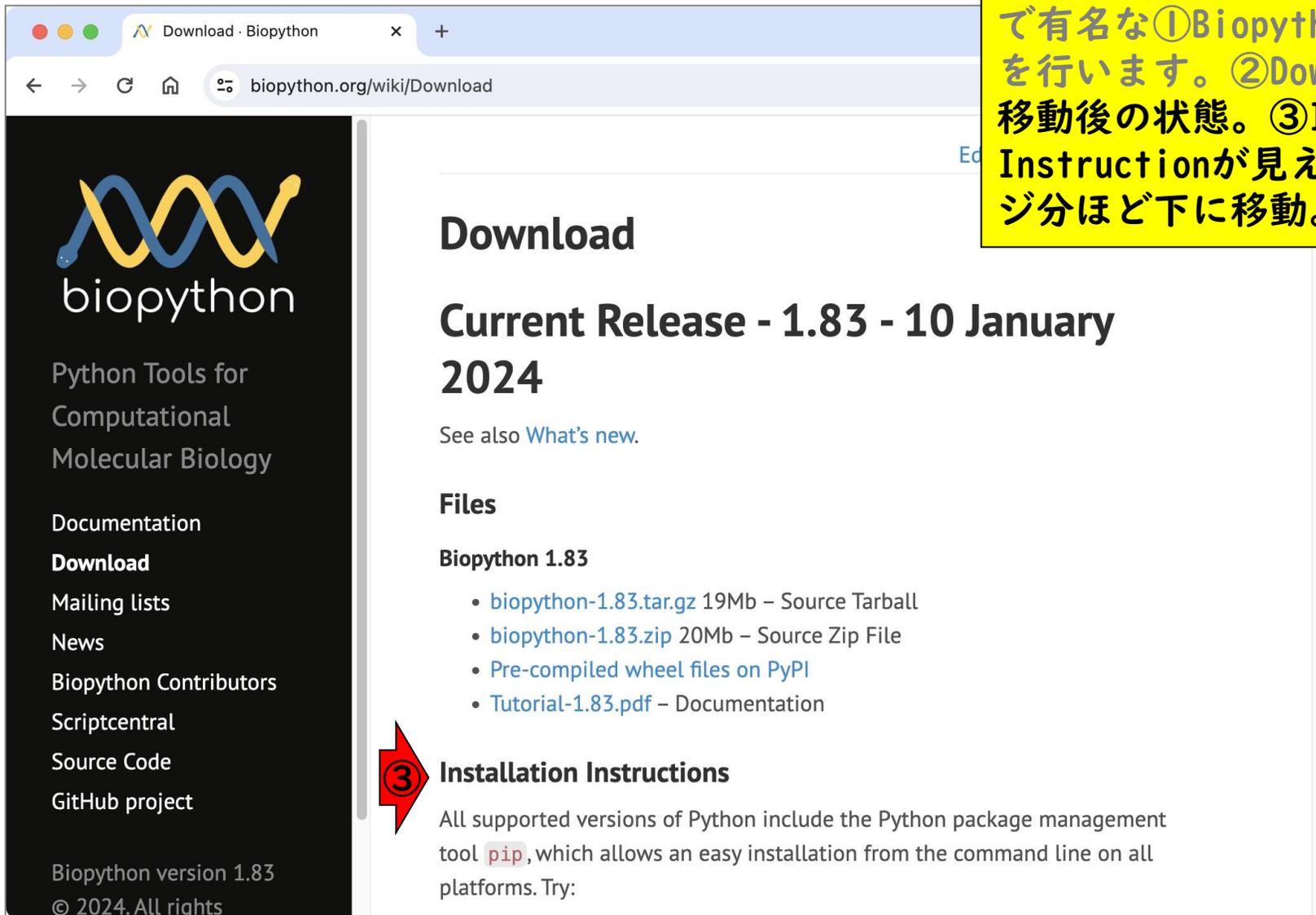
It is a distributed collaborative effort to develop Python libraries and applications which address the needs of current and future work in bioinformatics. The source code is made available under the [Biopython License](#), which is extremely liberal and compatible with almost every license in the world.

We are a member project of the [Open Bioinformatics Foundation \(OBF\)](#), who take care of our domain name and hosting for our mailing list etc. The OBF used to host our development repository, issue tracker and website but these are now on [GitHub](#).

This page will help you download and install Biopython, and start using the libraries and tools.

# パッケージのインストール2

Rとパッケージの関係同様、Pythonも様々なパッケージをインストールして利用します。ここでは、生命科学分野で有名な①Biopythonのインストールを行います。②Downloadをクリック。移動後の状態。③Installation Instructionが見えるように、半ページ分ほど下に移動。



Download · Biopython

biopython.org/wiki/Download

## Download

### Current Release - 1.83 - 10 January 2024

See also [What's new](#).

#### Files

##### Biopython 1.83

- [biopython-1.83.tar.gz](#) 19Mb – Source Tarball
- [biopython-1.83.zip](#) 20Mb – Source Zip File
- [Pre-compiled wheel files on PyPI](#)
- [Tutorial-1.83.pdf](#) – Documentation

#### ③ Installation Instructions

All supported versions of Python include the Python package management tool `pip`, which allows an easy installation from the command line on all platforms. Try:

# パッケージのインストール3

こんな感じ。インストールの基本形は、  
コマンドプロンプトで④を実行する  
だけです。つまり...

**③** Installation Instructions

All supported versions of Python include the Python package management tool `pip`, which allows an easy installation from the command line on all platforms. Try:

```
pip install biopython
```

**④**

For updating an older version of Biopython try:

```
pip install biopython --upgrade
```

This will remove older versions of Biopython and NumPy before it installs the recent versions.

Should you wish to uninstall Biopython:

```
pip uninstall biopython
```

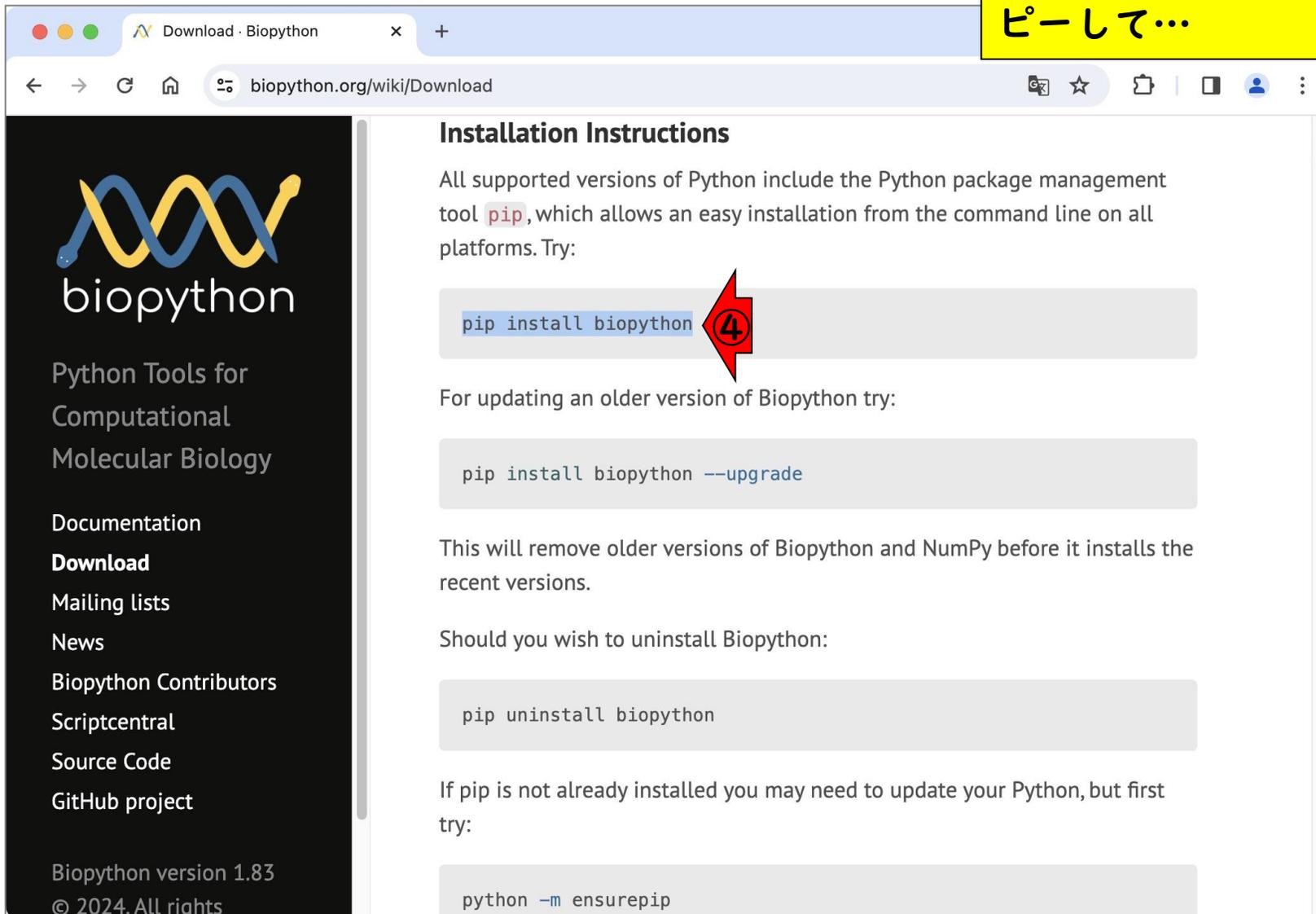
If pip is not already installed you may need to update your Python, but first try:

```
python -m ensurepip
```

biopython  
Python Tools for  
Computational  
Molecular Biology  
Documentation  
**Download**  
Mailing lists  
News  
Biopython Contributors  
Scriptcentral  
Source Code  
GitHub project  
Biopython version 1.83  
© 2024. All rights

# パッケージのインストール4

こんな感じ。インストールの基本形は、  
コマンドプロンプトで④を実行する  
だけです。つまり、④のコマンドをコ  
ピーして…



The screenshot shows a web browser window with the URL `biopython.org/wiki/Download`. The page title is "Installation Instructions". The main content describes how to install Biopython using `pip`. A red lightning bolt icon with the number 4 inside points to the command `pip install biopython` in a code block. Below this, instructions for updating and uninstalling are provided, with another code block showing `python -m ensurepip` for updating Python.

**Installation Instructions**

All supported versions of Python include the Python package management tool `pip`, which allows an easy installation from the command line on all platforms. Try:

```
pip install biopython
```

For updating an older version of Biopython try:

```
pip install biopython --upgrade
```

This will remove older versions of Biopython and NumPy before it installs the recent versions.

Should you wish to uninstall Biopython:

```
pip uninstall biopython
```

If `pip` is not already installed you may need to update your Python, but first try:

```
python -m ensurepip
```

biopython  
Python Tools for Computational Molecular Biology  
Documentation  
Download  
Mailing lists  
News  
Biopython Contributors  
Scriptcentral  
Source Code  
GitHub project  
Biopython version 1.83  
© 2024. All rights

# パッケージのインストール5

こんな感じ。インストールの基本形は、コマンドプロンプトで④を実行するだけです。つまり、④のコマンドをコピーして、⑤ターミナル上でペーストし、リターン。

student — -zsh — 88x28

Last login: Thu Mar 28 20:43:56 on ttys000

~ % 1 + 1

zsh: command not found: 1

~ % python3

Python 3.12.2 (v3.12.2:6abddd9f6a, Feb 6 2024, 17:02:06) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

[>>> 1 + 1

2

[>>> aa = 5

[>>> aa

5

[>>> quit()

~ %

~ % **pip install biopython**



# パッケージのインストール6

こんな感じ。インストールの基本形は、コマンドプロンプトで④を実行するだけです。つまり、④のコマンドをコピーして、⑤ターミナル上でペーストし、リターン。⑥残念ながらpipコマンドが無いというメッセージ。

```
student --zsh-- 88x28
Last login: Thu Mar 28 20:43:56 on ttys000
~ % 1 + 1
zsh: command not found: 1
~ % python3
Python 3.12.2 (v3.12.2:6abddd9f6a, Feb 6 2024, 17:02:06) [Clang 13.0.0 (clang-1300.0.29
.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> 1 + 1
2
[>>> aa = 5
[>>> aa
5
[>>> quit()
~ %
~ % pip install biopython
zsh: command not found: pip
~ %
```



# パッケージのインストール6

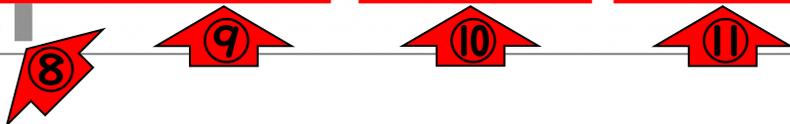
こんな感じ。インストールの基本形は、コマンドプロンプトで④を実行するだけです。つまり、④のコマンドをコピーして、⑤ターミナル上でペーストし、リターン。⑥残念ながらpipコマンドが無いというメッセージ。⑦pipをpip3に変更してリターン。

```
student --zsh -- 88x28
Last login: Thu Mar 28 20:43:56 on ttys000
~ % 1 + 1
zsh: command not found: 1
~ % python3
Python 3.12.2 (v3.12.2:6abddd9f6a, Feb  6 2024, 17:02:06) [Clang 13.0.0 (clang-1300.0.29
.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> 1 + 1
2
[>>> aa = 5
[>>> aa
5
[>>> quit()
~ %
~ % pip install biopython
zsh: command not found: pip
~ % pip3 install biopython
```

7

# パッケージのインストール7

```
student --zsh-- 88x28
~ % 1 + 1
zsh: command not found: 1
~ % python3
Python 3.12.2 (v3.12.2:6abddd9f6a, Feb 6 2024, 17:02:06) [Clang 13
.30] on darwin
Type "help", "copyright", "credits" or "license" for more informati
[>>> 1 + 1
2
[>>> aa = 5
[>>> aa
5
[>>> quit()
~ %
~ % pip install biopython
zsh: command not found: pip
~ % pip3 install biopython
Collecting biopython
  Downloading biopython-1.83-cp312-cp312-macosx_10_9_x86_64.whl.metadata (13 kB)
Collecting numpy (from biopython)
  Downloading numpy-1.26.4-cp312-cp312-macosx_10_9_x86_64.whl.metadata (61 kB)
----- 61.1/61.1 kB 1.3 MB/s eta 0:00:00
  Downloading biopython-1.83-cp312-cp312-macosx_10_9_x86_64.whl (2.7 MB)
----- 2.7/2.7 MB 3.1 MB/s eta 0:00:00
  Downloading numpy-1.26.4-cp312-cp312-macosx_10_9_x86_64.whl (20.3 MB)
----- 20.3/20.3 MB 38.6 MB/s eta 0:00:00
Installing collected packages: numpy, biopython
Successfully installed biopython-1.83 numpy-1.26.4
~ %
```



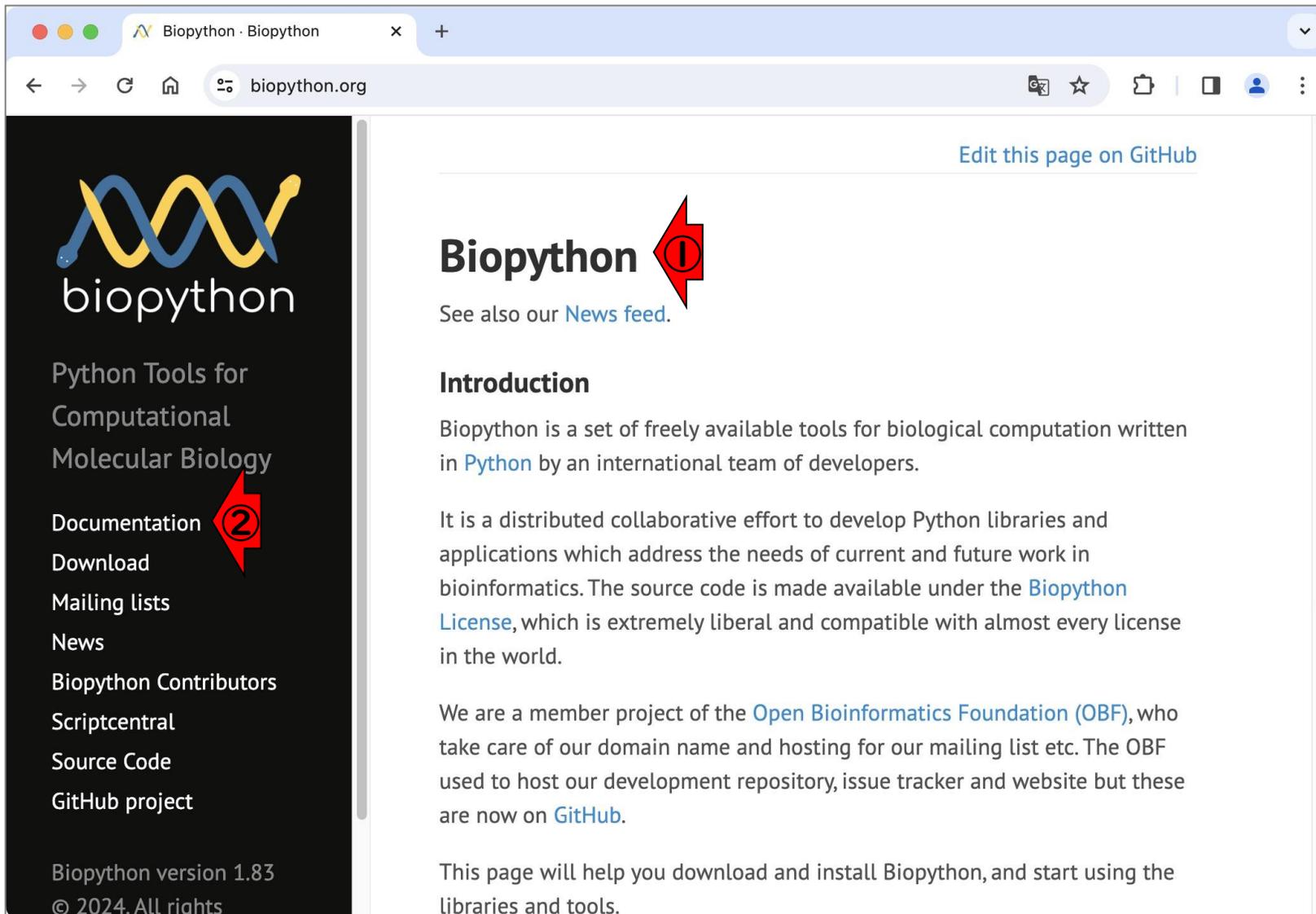
こんな感じ。インストールの基本形は、コマンドプロンプトで④を実行するだけです。つまり、④のコマンドをコピーして、⑤ターミナル上でペーストし、リターン。⑥残念ながらpipコマンドが無いというメッセージ。⑦pipをpip3に変更してリターン。⑧コマンドプロンプトが入力待ち状態になっている状態なので、インストール終了。⑨Successfully installedという記載内容からも、インストールは無事成功したことがわかります。⑩biopythonだけでなく、⑪numpyもインストールされました。この状態で一旦終了してもOKです。

# Contents

- Pythonのバージョン確認
- Python本体のインストール
- インストール確認
- Pythonの起動と終了(optional)
- 基本的な利用法(optional)
- パッケージのインストール(optional)
- Biopythonの基本的な利用法(optional)

# Biopythonの基本的な利用法 I

①Biopythonの基本的な利用法を紹介します。②Documentationをクリック。



The screenshot shows the Biopython website homepage. The browser address bar displays 'biopython.org'. The page features a dark sidebar on the left with the Biopython logo and navigation links. The main content area includes a 'Biopython' heading with a red arrow pointing to it, a 'News feed' link, an 'Introduction' section, and a 'Documentation' link in the sidebar with a red arrow and the number '2' pointing to it. The page also includes a 'GitHub' link and a footer with version and copyright information.

Python Tools for Computational Molecular Biology

Documentation ②

Download

Mailing lists

News

Biopython Contributors

Scriptcentral

Source Code

GitHub project

Biopython version 1.83  
© 2024. All rights

Edit this page on GitHub

## Biopython

See also our [News feed](#).

### Introduction

Biopython is a set of freely available tools for biological computation written in [Python](#) by an international team of developers.

It is a distributed collaborative effort to develop Python libraries and applications which address the needs of current and future work in bioinformatics. The source code is made available under the [Biopython License](#), which is extremely liberal and compatible with almost every license in the world.

We are a member project of the [Open Bioinformatics Foundation \(OBF\)](#), who take care of our domain name and hosting for our mailing list etc. The OBF used to host our development repository, issue tracker and website but these are now on [GitHub](#).

This page will help you download and install Biopython, and start using the libraries and tools.

# Biopythonの基本的な利用法2

①Biopythonの基本的な利用法を紹介します。②Documentationをクリック。ググれば様々な情報が得られますが、例えば文字列を塩基配列やアミノ酸配列(Sequence)として取り扱ってくれる③Seq関数の利用例を見えます。③をクリック。

Documentation · Biopython

biopython.org/wiki/Documentation

## Documentation

New to Biopython? Check out the [Getting Started](#) page, or follow one of the links below.

- The **Biopython Tutorial and Cookbook** contains the bulk of Biopython documentation. It provides information to get you started with Biopython, in addition to specific documentation on a number of modules

[HTML](#), [PDF](#)

- API documentation for Biopython modules is generated directly from source code comments Sphinx autodoc:  
[Biopython API \(latest release\)](#), and [Biopython API \(in-development\)](#).

- [Wiki documentation](#)
  - ③** [Seq](#) and [SeqRecord](#) objects
    - [Bio.SeqIO](#) - sequence input/output
    - [Bio.AlignIO](#) - alignment input/output
    - [Bio.PopGen](#) - population genetics
    - [Bio.PDB](#) - structural bioinformatics

Python Tools for Computational Molecular Biology

Documentation  
Download  
Mailing lists  
News  
Biopython Contributors  
Scriptcentral  
Source Code  
GitHub project

Biopython version 1.83  
© 2024. All rights

# Biopythonの基本的な利用法3

④ 「Biopythonでは配列を`Seq`オブジェクトとして取り扱う」的なことが書かれています。⑤が見えるように、半ページ分ほど下に移動。

Handling sequences with the `Seq` class.

In Biopython, sequences are usually held as `Seq` objects, which add various biological methods on top of string like behaviour.

This page describes the Biopython `Seq` object, defined in the `Bio.Seq` module (together with related objects like the `MutableSeq`, plus some general purpose sequence functions). In addition to this wiki page, there is a whole chapter in the [Tutorial \(PDF\)](#) on the `Seq` object - plus its [API documentation](#) (which you can read online, or from within Python with the help command).

If you need to store additional information like a sequence identifier or name, or even more details like a description or annotation, then we use a `SeqRecord` object instead. These are the sequence records used by the `SeqIO` module for reading and writing sequence files.

## ⑤ The Seq Object



Python Tools for  
Computational  
Molecular Biology

Documentation

Download

Mailing lists

News

Biopython Contributors

Scriptcentral

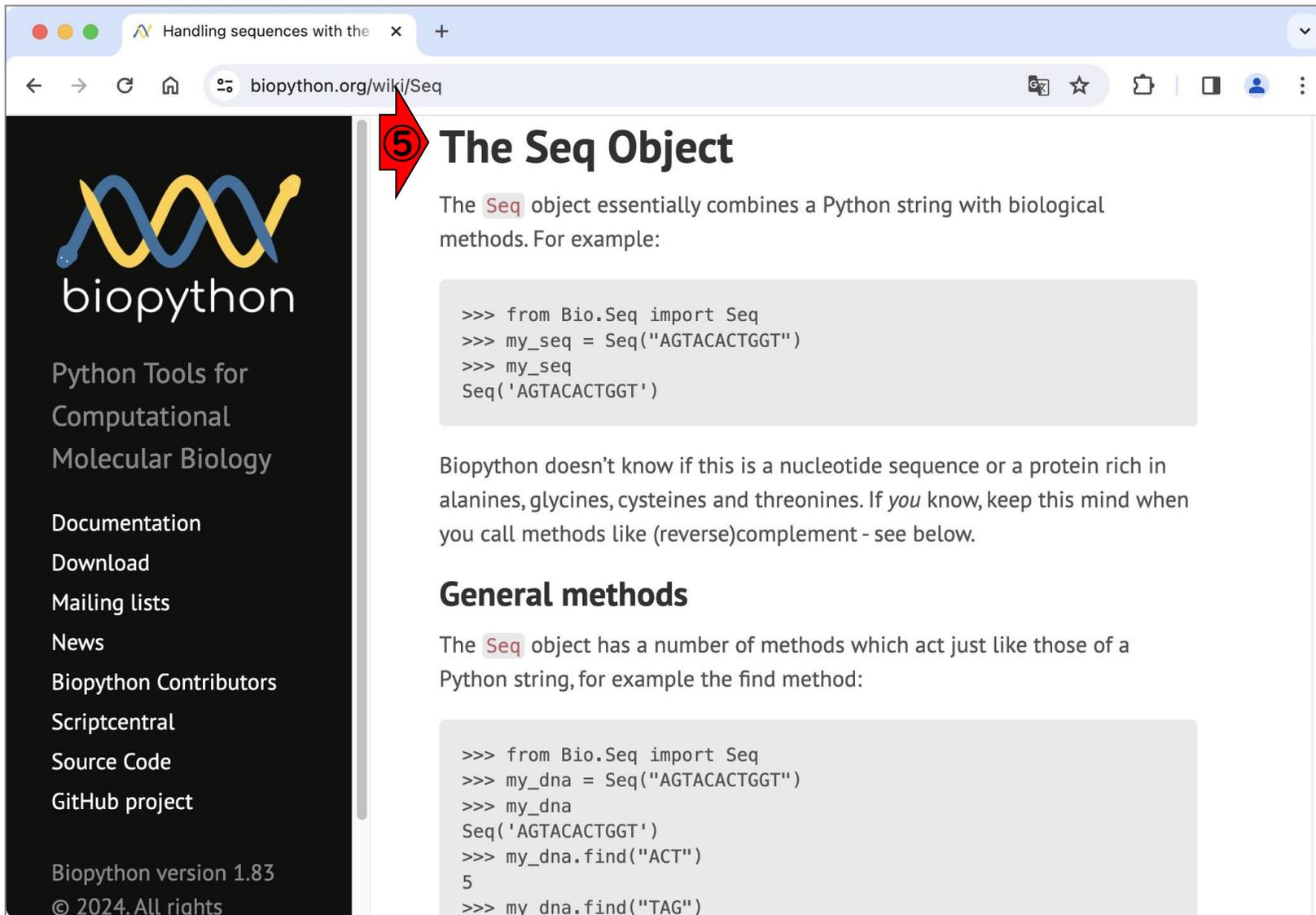
Source Code

GitHub project

Biopython version 1.83

© 2024. All rights

# Biopythonの基本的な利用法4



The screenshot shows a web browser window with the URL `biopython.org/wiki/Seq`. The page title is "The Seq Object". A red arrow with the number 5 points to the title. The page content includes a description of the `Seq` object, a code block showing how to create a `Seq` object, and a section for "General methods" with another code block showing the `find` method.

**5 The Seq Object**

The `Seq` object essentially combines a Python string with biological methods. For example:

```
>>> from Bio.Seq import Seq
>>> my_seq = Seq("AGTACACTGGT")
>>> my_seq
Seq('AGTACACTGGT')
```

Biopython doesn't know if this is a nucleotide sequence or a protein rich in alanines, glycines, cysteines and threonines. If *you* know, keep this mind when you call methods like `(reverse)complement` - see below.

### General methods

The `Seq` object has a number of methods which act just like those of a Python string, for example the `find` method:

```
>>> from Bio.Seq import Seq
>>> my_dna = Seq("AGTACACTGGT")
>>> my_dna
Seq('AGTACACTGGT')
>>> my_dna.find("ACT")
5
>>> my_dna.find("TAG")
```

# Biopythonの基本的な利用法5

こんな感じ。⑥のあたりを眺めて、「⑥の3行は、コマンドプロンプトでpython3と打ち込んでPython環境にしてから打ち込むコマンド。そして最初の>>>は実際には打ち込まない。」と解釈します。



Python Tools for  
Computational  
Molecular Biology

Documentation

Download

Mailing lists

News

Biopython Contributors

Scriptcentral

Source Code

GitHub project

Biopython version 1.83

© 2024. All rights

## The Seq Object

The `Seq` object essentially combines a Python string with biological methods. For example:



```
>>> from Bio.Seq import Seq
>>> my_seq = Seq("AGTACACTGGT")
>>> my_seq
Seq('AGTACACTGGT')
```

Biopython doesn't know if this is a nucleotide sequence or a protein rich in alanines, glycines, cysteines and threonines. If *you* know, keep this mind when you call methods like `(reverse)complement` - see below.

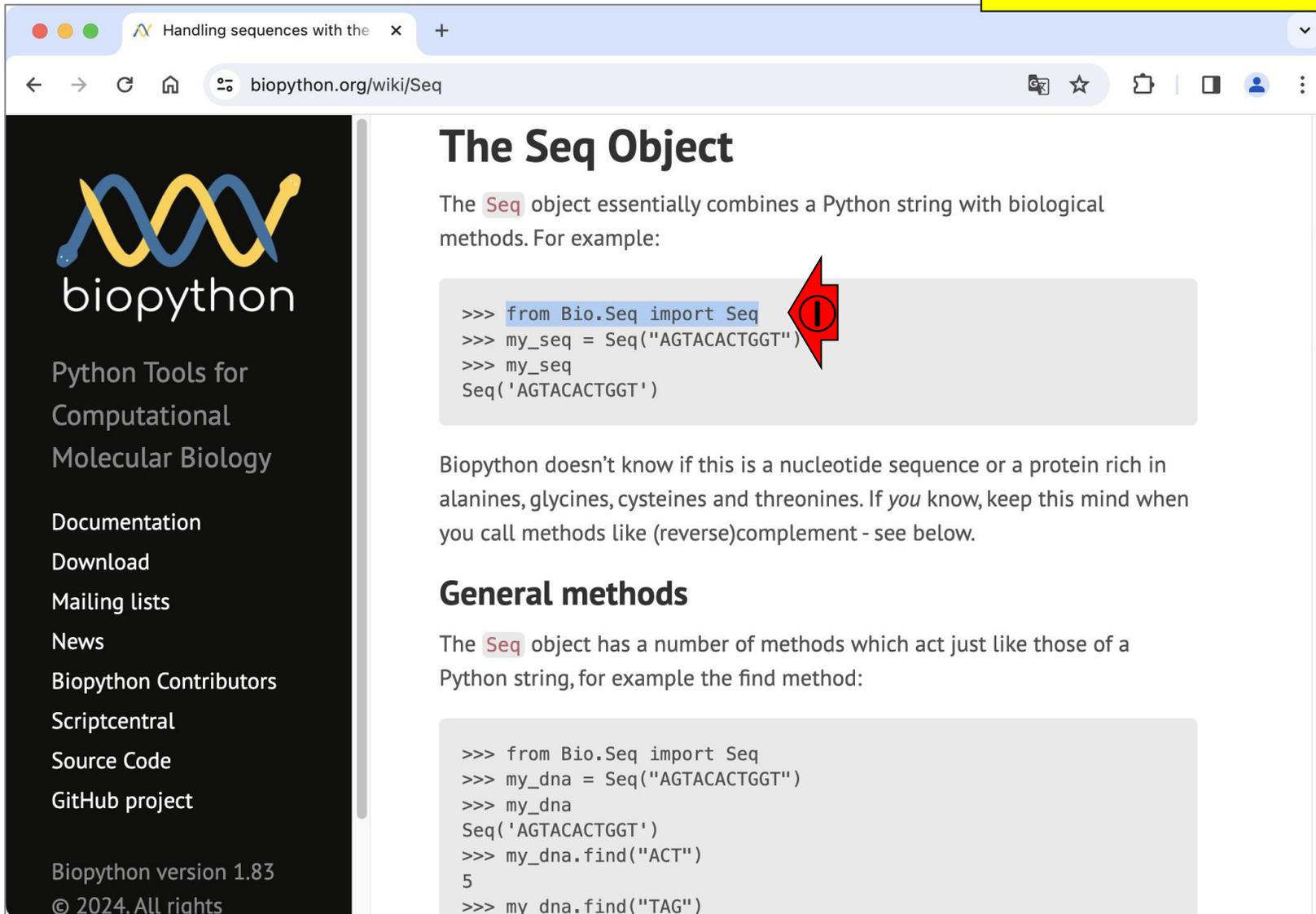
## General methods

The `Seq` object has a number of methods which act just like those of a Python string, for example the `find` method:

```
>>> from Bio.Seq import Seq
>>> my_dna = Seq("AGTACACTGGT")
>>> my_dna
Seq('AGTACACTGGT')
>>> my_dna.find("ACT")
5
>>> my_dna.find("TAG")
```

# Biopythonの基本的な利用法6

まずは、BiopythonのSeq関数を利用できるようにロードすべく、①反転部分をコピー。



The screenshot shows the Biopython website's documentation for the Seq object. The left sidebar contains the Biopython logo and navigation links: Python Tools for Computational Molecular Biology, Documentation, Download, Mailing lists, News, Biopython Contributors, Scriptcentral, Source Code, and GitHub project. The main content area is titled 'The Seq Object' and explains that the Seq object combines a Python string with biological methods. A code block shows the import and creation of a Seq object: 

```
>>> from Bio.Seq import Seq
>>> my_seq = Seq("AGTACACTGGT")
>>> my_seq
Seq('AGTACACTGGT')
```

 A red arrow points to the `from Bio.Seq import Seq` line. Below the code block, it notes that Biopython doesn't know if the sequence is nucleotide or protein-rich. Another code block shows the `find` method: 

```
>>> from Bio.Seq import Seq
>>> my_dna = Seq("AGTACACTGGT")
>>> my_dna
Seq('AGTACACTGGT')
>>> my_dna.find("ACT")
5
>>> my_dna.find("TAG")
```

# Biopythonの基本的な利用法7

まずは、BiopythonのSeq関数を利用できるようにロードすべく、①反転部分をコピー。ターミナルで②python3と打ち込んでPython環境(つまり③>>>が見える状態)にした状態で…

student — Python — 88x28

Last login: ② Mar 28 21:13:34 on ttys000

~ % python3

Python 3.12.2 (v3.12.2:6abddd9f6a, Feb 6 2024, 17:02:06) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

>>> ③

# Biopythonの基本的な利用法8

まずは、BiopythonのSeq関数を利用できるようにロードすべく、①反転部分をコピー。ターミナルで②python3と打ち込んでPython環境(つまり③>>>が見える状態)にした状態で、④ペーストしてリターン。特にエラーが出ていないことから、BiopythonのSeq関数が利用できるようになりました。

student — Python — 88x28

Last login: Thu Mar 28 21:13:34 on ttys000

~ % python3

Python 3.12.2 (v3.12.2:6abddd9f6a, Feb 6 2024, 17:02:06) [Clang 13.0.0] on darwin

Type "help", "copyright", "credits" or "license" for more information

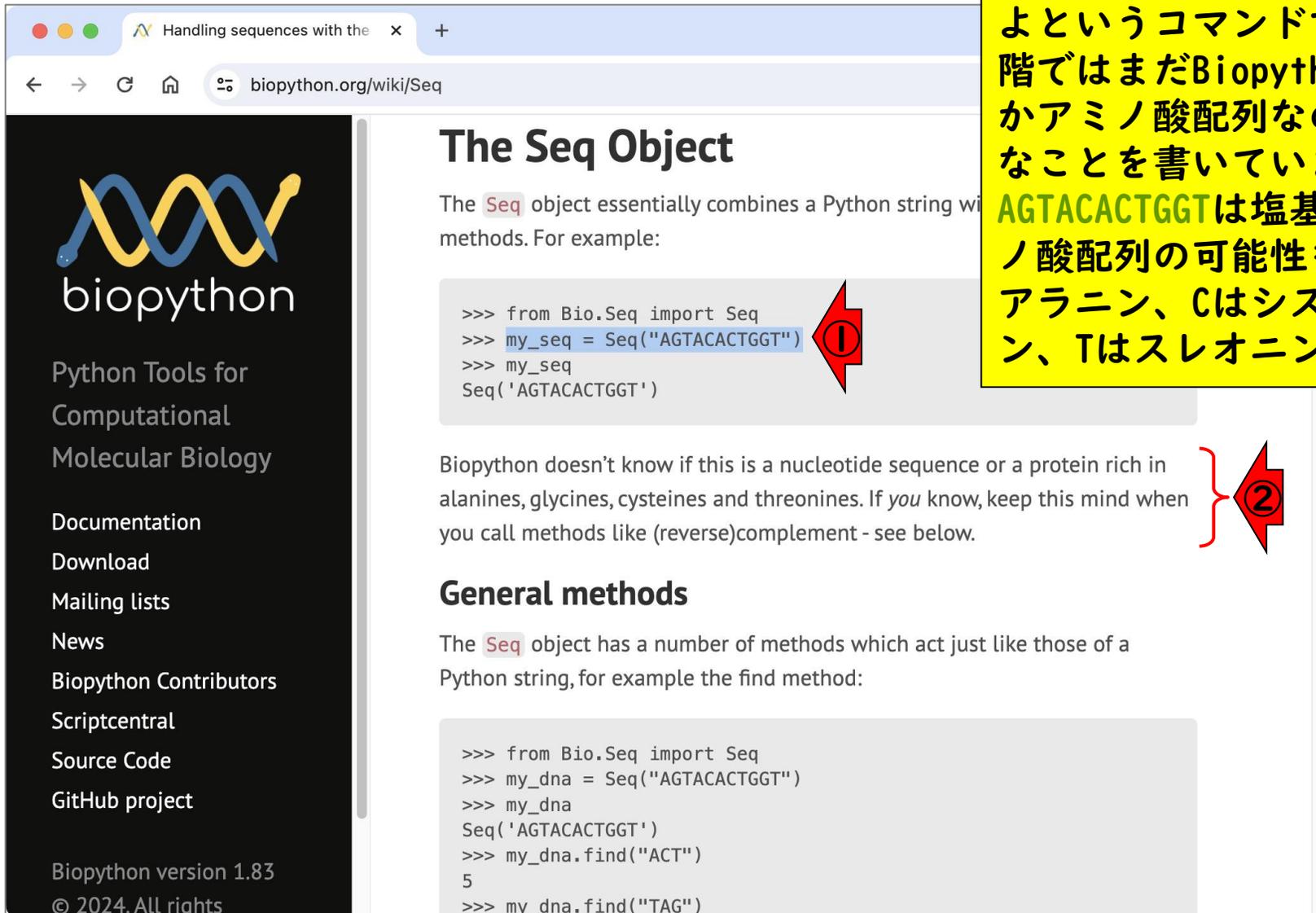
>>> from Bio.Seq import Seq

>>> █



# Biopythonの基本的な利用法9

①は、AGTACACTGGTという文字列を、Seq関数で読み込むことで、`Seq`オブジェクトにした状態でmy\_seqに格納せよというコマンドです。②は、この段階ではまだBiopythonは塩基配列なのかアミノ酸配列なのかはわからない的なことを書いています。直感的にはAGTACACTGGTは塩基配列ですが、アミノ酸配列の可能性もあるからです(Aはアラニン、Cはシステイン、Gはグリシン、Tはスレオニン)。



Handling sequences with the

biopython.org/wiki/Seq

## The Seq Object

The `Seq` object essentially combines a Python string with methods. For example:

```
>>> from Bio.Seq import Seq
>>> my_seq = Seq("AGTACACTGGT")
>>> my_seq
Seq('AGTACACTGGT')
```

Biopython doesn't know if this is a nucleotide sequence or a protein rich in alanines, glycines, cysteines and threonines. If you know, keep this mind when you call methods like (reverse)complement - see below.

### General methods

The `Seq` object has a number of methods which act just like those of a Python string, for example the find method:

```
>>> from Bio.Seq import Seq
>>> my_dna = Seq("AGTACACTGGT")
>>> my_dna
Seq('AGTACACTGGT')
>>> my_dna.find("ACT")
5
>>> my_dna.find("TAG")
```



Python Tools for  
Computational  
Molecular Biology

Documentation

Download

Mailing lists

News

Biopython Contributors

Scriptcentral

Source Code

GitHub project

Biopython version 1.83

© 2024. All rights

# Biopythonの基本的な…10

student — Python — 88x28

```
Last login: Thu Mar 28 21:13:34 on ttys000
~ % python3
Python 3.12.2 (v3.12.2:6abddd9f6a, Feb  6 2024, 17:02:06) [Clang 13
.30] on darwin
Type "help", "copyright", "credits" or "license" for more informati
[>>> from Bio.Seq import Seq
[>>> my_seq = Seq("AGTACACTGGT")
[>>> my_seq
Seq('AGTACACTGGT')
>>> █
```

①は、AGTACACTGGTという文字列を、Seq関数で読み込むことで、`Seq`オブジェクトにした状態でmy\_seqに格納せよというコマンドです。②は、この段階ではまだBiopythonは塩基配列なのかアミノ酸配列なのかはわからない的なことを書いています。直感的にはAGTACACTGGTは塩基配列ですが、アミノ酸配列の可能性もあるからです(Aはアラニン、Cはシステイン、Gはグリシン、Tはスレオニン)。さきほどの①とそのすぐ下のコマンドを実行した結果。赤枠内が実際にコピペ実行したものになります。

# Biopythonの基本的な…II

ここまでで、①灰色部分全体と、②実際に打ち込んだ赤枠内との関係がわかりました。このノリで、③以降を解説を見ながら順を追って理解していくとよいと思います。

## The Seq Object

The `Seq` object essentially combines a Python string with biological methods. For example:

```
>>> from Bio.Seq import Seq
>>> my_seq = Seq("AGTACACTGGT")
>>> my_seq
Seq('AGTACACTGGT')
```

Biopython doesn't know if this is a nucleotide sequence or a protein rich in alanines, glycines, cysteines and threonines. If *you* know, keep this mind when you call methods like `(reverse)complement` - see below.

### General methods

The `Seq` object has a number of methods which act just like those of a Python string, for example the `find` method:

```
>>> from Bio.Seq import Seq
>>> my_dna = Seq("AGTACACTGGT")
>>> my_dna
Seq('AGTACACTGGT')
>>> my_dna.find("ACT")
5
>>> my_dna.find("TAG")
```



Python Tools for  
Computational  
Molecular Biology

Documentation

Download

Mailing lists

News

Biopython Contributors

Scriptcentral

Source Code

GitHub project

Biopython version 1.83

© 2024. All rights