

Python 3.8 のインストール（Windows上）

サイト 構成	データベース関 連技術	データの 扱い	インストール、設 定、利用	プログラ ミング	授業 資料	サポート ページ	連絡先、業 績など
-----------	----------------	------------	------------------	-------------	----------	-------------	--------------

Windows での Python 3.8 のインストール、インストール後の設定、Python の各種パッケージのインストール、性能確認の手順をスクリーンショット等で説明する。

※ **TensorFlow を使う予定がある場合**は、<https://pypi.org/project/tensorflow-gpu/#files>で、必要な Python のバージョンを確認しておくこと。2020/5 時点では、Python 3.5 か 3.6 か 3.7 か 3.8

【このページの目次】

1. [Windows 用の Python のダウンロードとインストール](#)
2. [Python のインストール後の確認](#)
3. [Python のインストール後の設定](#)
4. [性能の確認](#)

参考 Web ページ: <https://docs.python.org/ja/3/using/windows.html>

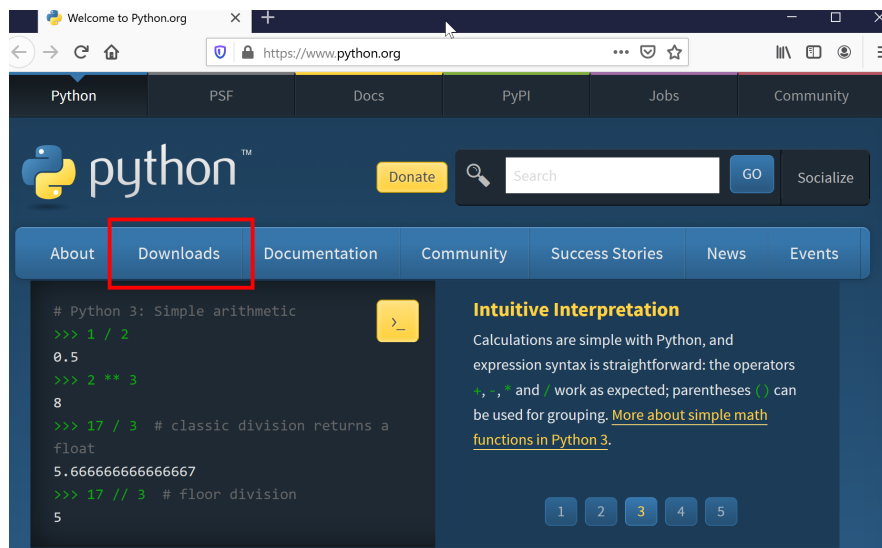
Windows 用の Python のダウンロードとインストール

1. Python の Web ページを開く

<http://www.python.org/>を開く

2. ダウンロード用ページへジャンプ

ページの上の方にある「**Downloads**」をクリック



3. Python のバージョンを選ぶ

Looking for a specific release?
Python releases by version number:

Release version	Release date	Click for more
Python 3.8.3	May 13, 2020	Download Release Notes
Python 3.8.3rc1	April 29, 2020	Download Release Notes
Python 2.7.18	April 20, 2020	Download Release Notes
Python 3.7.7	March 10, 2020	Download Release Notes
Python 3.8.2	Feb. 24, 2020	Download Release Notes
Python 3.8.1	Dec. 18, 2019	Download Release Notes
Python 3.7.6	Dec. 18, 2019	Download Release Notes

[View older releases](#)

TensorFlow を使う予定がある場合は、<https://pypi.org/project/tensorflow-gpu/#files>で、必要な Python のバージョンを確認しておくこと。2020/5 時点では、Python 3.5 か 3.6 か 3.7 か 3.8

以下、**Python 3.8.2 を選んだ**として説明を続ける。他のバージョンでも以下の手順はだいたい同じである。

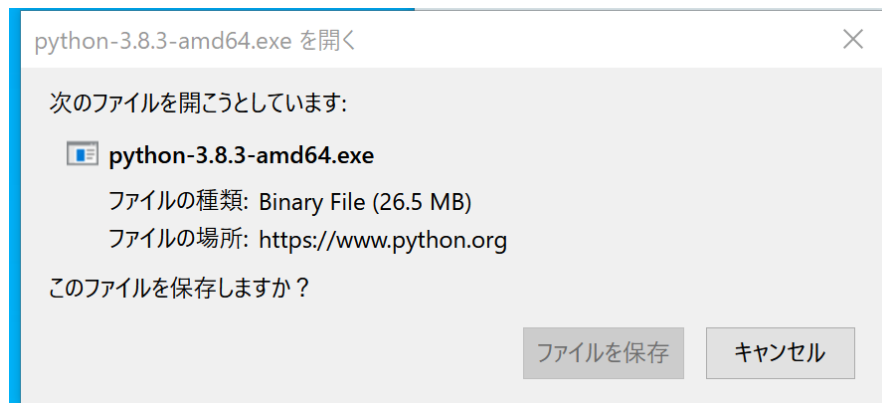
4. 画面が切り替わる。ファイルの種類を選ぶ。

Windows の 64ビット版のインストーラをダウンロードしたいので、「**x86_64-executable-installer**」を選ぶ

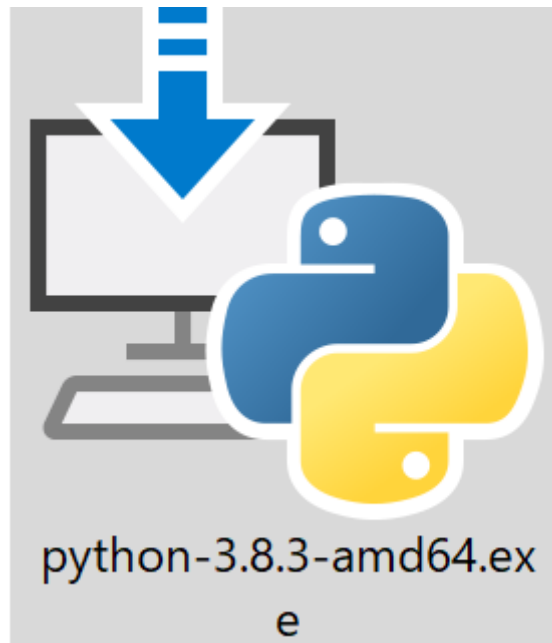
Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		a7c10a2ac9d62de75a0ca5204e2e7d07	24067487	SIG
XZ compressed source tarball	Source release		3000cf50aaa113052ae82fd2122ca78	17912964	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	dd5e7f64e255d21f8d407f39a7a41ba9	30119781	SIG
Windows help file	Windows		4aeeebd7cc8dd90d61e7cfd9a9cb9422	8568303	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	c12ffe7f4c1b447241d5d2aedc9b5d01	8175801	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	fd2458fa0e9ead1dd9fbc2370a42853b	27805800	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	17e989d2fec7f7913cf987825b695c4	1364136	SIG
Windows x86 embeddable zip file	Windows		8ee09403ec0cc2e89d43b4a4fd1521e	7330315	SIG
Windows x86 executable installer	Windows		452373e2c467c14220efeb10f40c231f	26744744	SIG
Windows x86 web-based installer	Windows		fe72582bca3dbe07451fd05ece1d752	1325800	SIG

5. ダウンロードが始まる



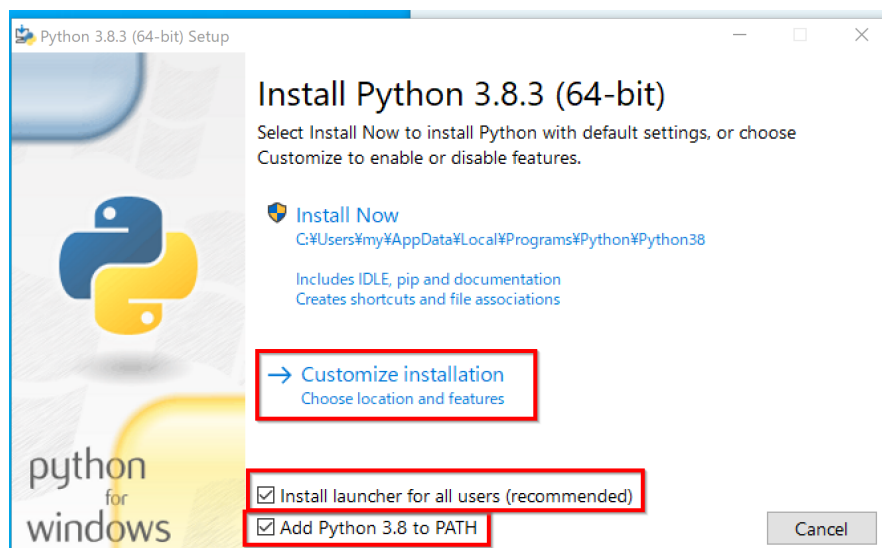
6. いまダウンロードしたファイルを実行する



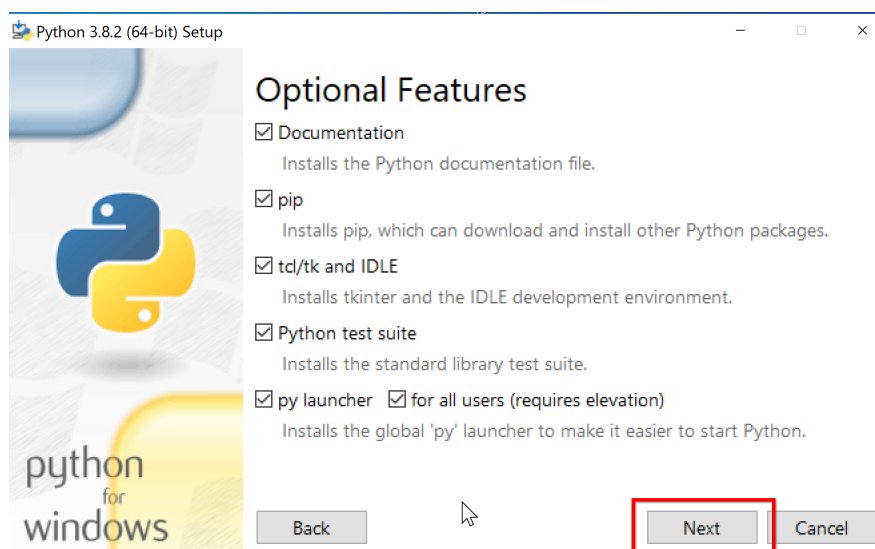
7. Pythonランチャーをインストールするために、「**Install launcher for all users (recommended)**」をチェック。

そして、「**Add Python 3.8 to PATH**」をチェック。そして、「**Customize Installation**」をクリック

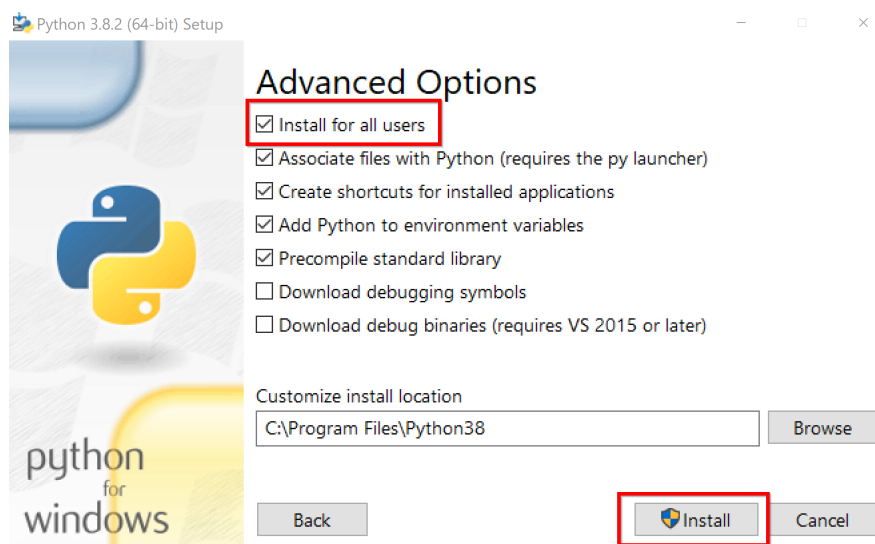
※ すでに Python ランチャーをインストール済みのときは、「**Install launcher for all users (recommended)**」をチェックしない。



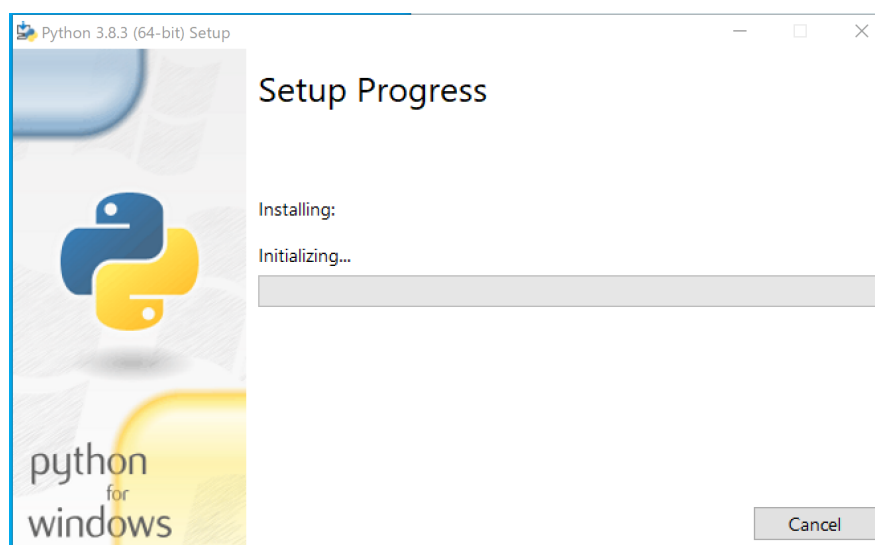
8. オプションの機能は、既定（デフォルト）のままでよい。 「**Next**」をクリック



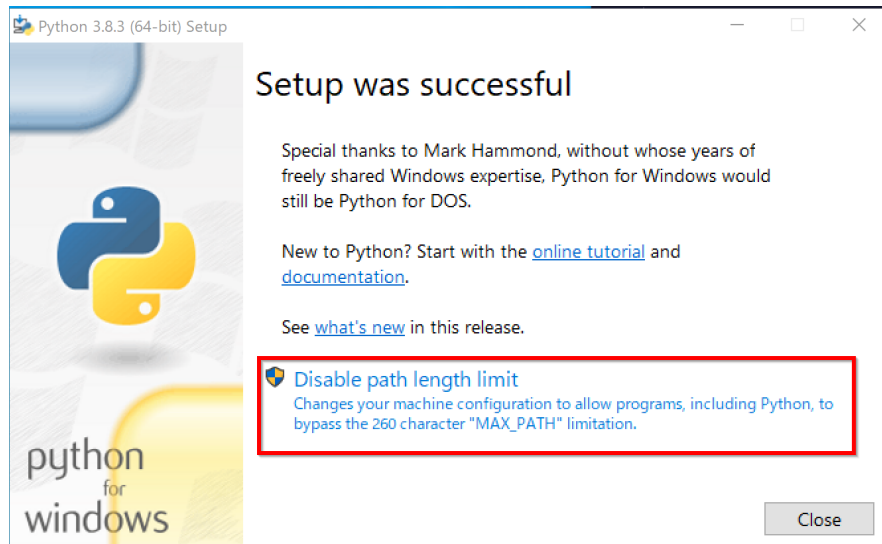
9. 「**Install for all users**」を選んでおいたほうが、複数人で使えて便利。 「**Install**」をクリック



10. インストールが始まる

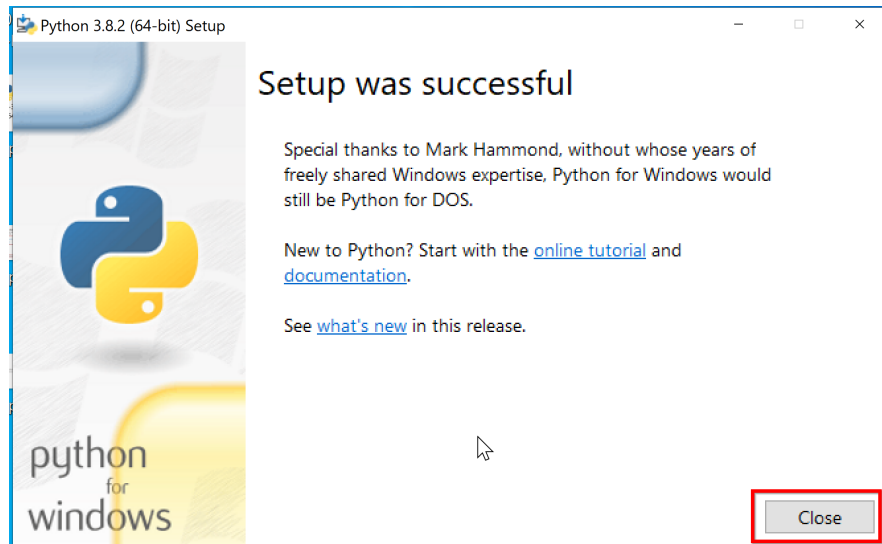


11. 「**Disable path length limit**」が表示される場合がある。クリックして、パス長の制限を解除する



12. インストールの終了

「Close」をクリック



Python のインストール後の確認

1. Windowsのユーザ環境変数PATHの **先頭**部分を確認

さきほど、「**Add Python ... to PATH**」をチェックしたので、Python についての設定が自動で行われたことを確認する

環境変数名の編集

```
C:\Program Files\Python38\Scripts\
C:\Program Files\Python38\
```

2. Windows で、コマンドプロンプトを実行.

3. **py** と **pip** にパスが通っていることの確認

次のコマンドを実行

```
where py
where pip
```

※ 表示は下図と違うことがありえる。エラーメッセージが**出ない**ことを確認。

```
C:\Users\my>where py
C:\Windows\py.exe

C:\Users\my>where pip
C:\Program Files\Python38\Scripts\pip.exe

C:\Users\my>
```

4. python のバージョンの確認

※ エラーメッセージが**出ない**ことを確認。

```
python --version
```

```
C:\Users\my>python --version
Python 3.8.3

C:\Users\my>
```

5. Python のビルドに用いられたコンパイラのバージョン番号の確認

```
python
```

下の実行例では、バージョン番号として「**1924**」が表示されている

```
C:\Users\my>python
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

6. 引き続き、次のPythonプログラムを実行し、バージョン番号を確認する

下の実行例では、バージョン番号として「**14.1**」が表示されている

```
from distutils.msvc9compiler import *
get_build_version()
```

```
C:\Users\my>python
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from distutils.msvc9compiler import *
>>> get_build_version()
14.2
>>>
```

7. exit() で終了

```
C:\Users\my>python
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from distutils.msvc9compiler import *
>>> get_build_version()
14.2
>>> exit()

C:\Users\my>
```

8. pip の動作確認

Python のパッケージも同時にインストールされることが分かる。

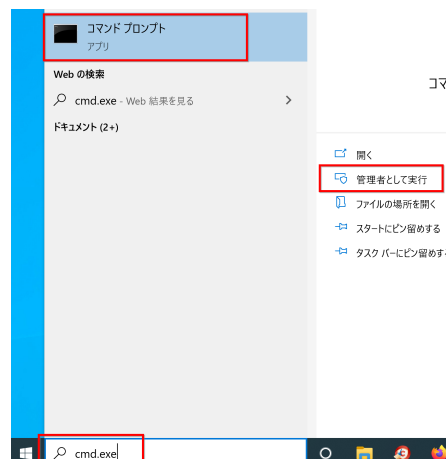
※ エラーメッセージが**出ない**ことを確認。

```
pip list
```

```
C:\Users\my>pip list
Package      Version
-----
pip          19.2.3
setuptools   41.2.0
WARNING: You are using pip version 19.2.3, however version 20.1.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\Users\my>
```

Python のインストール後の設定

1. Windows で、コマンドプロンプトを**管理者として実行**。



2. pip の**更新**

```
python -m pip install -U pip setuptools
```

```
管理者: コマンド プロンプト
Microsoft Windows [Version 10.0.20150.1000]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>python -m pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/43/84/23ed6a17
/pip-20.1.1-py2.py3-none-any.whl (1.5MB)
    | 1.5MB 1.7MB/s
Installing collected packages: pip
  Found existing installation: pip 19.2.3
  Uninstalling pip-19.2.3:
    Successfully uninstalled pip-19.2.3
  Successfully installed pip-20.1.1

C:\WINDOWS\system32>
```

3. **setuptools の更新**

```
python -m pip install -U setuptools
```

```
C:\WINDOWS\system32>python -m pip install -U setuptools
Collecting setuptools
  Downloading setuptools-47.3.1-py3-none-any.whl (582 kB)
    | 582 kB 2.2 MB/s
Installing collected packages: setuptools
  Attempting uninstall: setuptools
    Found existing installation: setuptools 41.2.0
    Uninstalling setuptools-41.2.0:
      Successfully uninstalled setuptools-41.2.0
  Successfully installed setuptools-47.3.1
C:\WINDOWS\system32>
```

4. 再度、**pip** の動作確認

※ エラーメッセージが**出ない**ことを確認.

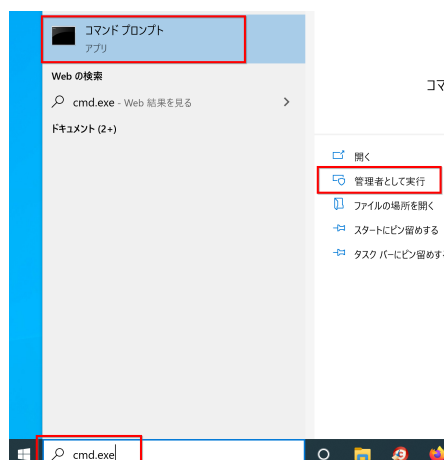
```
pip list
```

```
C:\WINDOWS\system32>pip list
Package      Version
-----
pip          20.1.1
setuptools   47.3.1
C:\WINDOWS\system32>
```

性能の確認

行列の積, 主成分分析, SVD, k-Means クラスタリングを実行し, 性能を確認する.

1. Windows で, コマンドプロンプトを管理者として実行.



2. まず, 前準備として, 次のコマンドを実行

```
python -m pip install -U numpy
python -m pip install -U scikit-learn
```



```
管理者: コマンド プロンプト
Microsoft Windows [Version 10.0.18363.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>pip install -U numpy
Collecting numpy
  Using cached numpy-1.18.4-cp38-cp38-win_amd64.whl (12.8 MB)
Installing collected packages: numpy
Successfully installed numpy-1.18.4

C:\WINDOWS\system32>pip install -U scikit-learn
Collecting scikit-learn
  Using cached scikit_learn-0.22.2.post1-cp38-cp38-win_amd64.whl (6.6 MB)
Collecting scipy>=0.17.0
  Using cached scipy-1.4.1-cp38-cp38-win_amd64.whl (31.0 MB)
Requirement already satisfied, skipping upgrade: numpy>=1.11.0 in c:\program file
s\python38\lib\site-packages (from scikit-learn) (1.18.4)
Collecting joblib>=0.11
  Using cached joblib-0.14.1-py2.py3-none-any.whl (294 kB)
Could not build wheels for numpy, since package 'wheel' is not installed.
Installing collected packages: scipy, joblib, scikit-learn
Successfully installed joblib-0.14.1 scikit-learn-0.22.2.post1 scipy-1.4.1

C:\WINDOWS\system32>
```

3. 性能の確認のため、次の Python プログラムを実行

```
import time
import numpy
import numpy.linalg
import sklearn.decomposition
import sklearn.cluster
X = numpy.random.rand(2000, 2000)
Y = numpy.random.rand(2000, 2000)
# 行列の積
a = time.time(); Z = numpy.dot(X, Y); print(time.time() - a)
# 主成分分析
pca = sklearn.decomposition.PCA(n_components = 2)
a = time.time(); pca.fit(X); X_pca = pca.transform(X);
print(time.time() - a)
# SVD
a = time.time(); U, S, V = numpy.linalg.svd(X); print(time.time() - a)
# k-means
a = time.time();
kmeans_model = sklearn.cluster.KMeans(n_clusters=10,
random_state=10).fit(X)
labels = kmeans_model.labels_
print(time.time() - a)
```

実行結果の例

- 行列の積: 0.19 秒
- 主成分分析: 0.13 秒
- SVD: 3.98 秒
- k-Means クラスタリング: 4.04 秒

```
C:\WINDOWS\system32>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import time
>>> import numpy
>>> import numpy.linalg
>>> import sklearn.decomposition
>>> import sklearn.cluster
>>> X = numpy.random.rand(2000, 2000)
>>> Y = numpy.random.rand(2000, 2000)
>>> # 行列の積
>>> a = time.time(); Z = numpy.dot(X, Y); print(time.time() - a)
0.1894090175628662
>>> # 主成分分析
>>> pca = sklearn.decomposition.PCA(n_components=2)
>>> a = time.time(); pca.fit(X); X_pca = pca.transform(X); print(time.time() - a)
PCA(copy=True, iterated_power='auto', n_components=2, random_state=None,
    svd_solver='auto', tol=0.0, whiten=False)
0.12841224670410156
>>> # SVD
>>> a = time.time(); U, S, V = numpy.linalg.svd(X); print(time.time() - a)
3.977161169052124
>>> # k-means
>>> a = time.time();
>>> kmeans_model = sklearn.cluster.KMeans(n_clusters=10, random_state=10).fit(X)
>>> labels = kmeans_model.labels_
>>> print(time.time() - a)
4.035630702972412
>>> exit()

C:\WINDOWS\system32>
```