pipenv system
GPU利用版

First ed. October 07, 2021

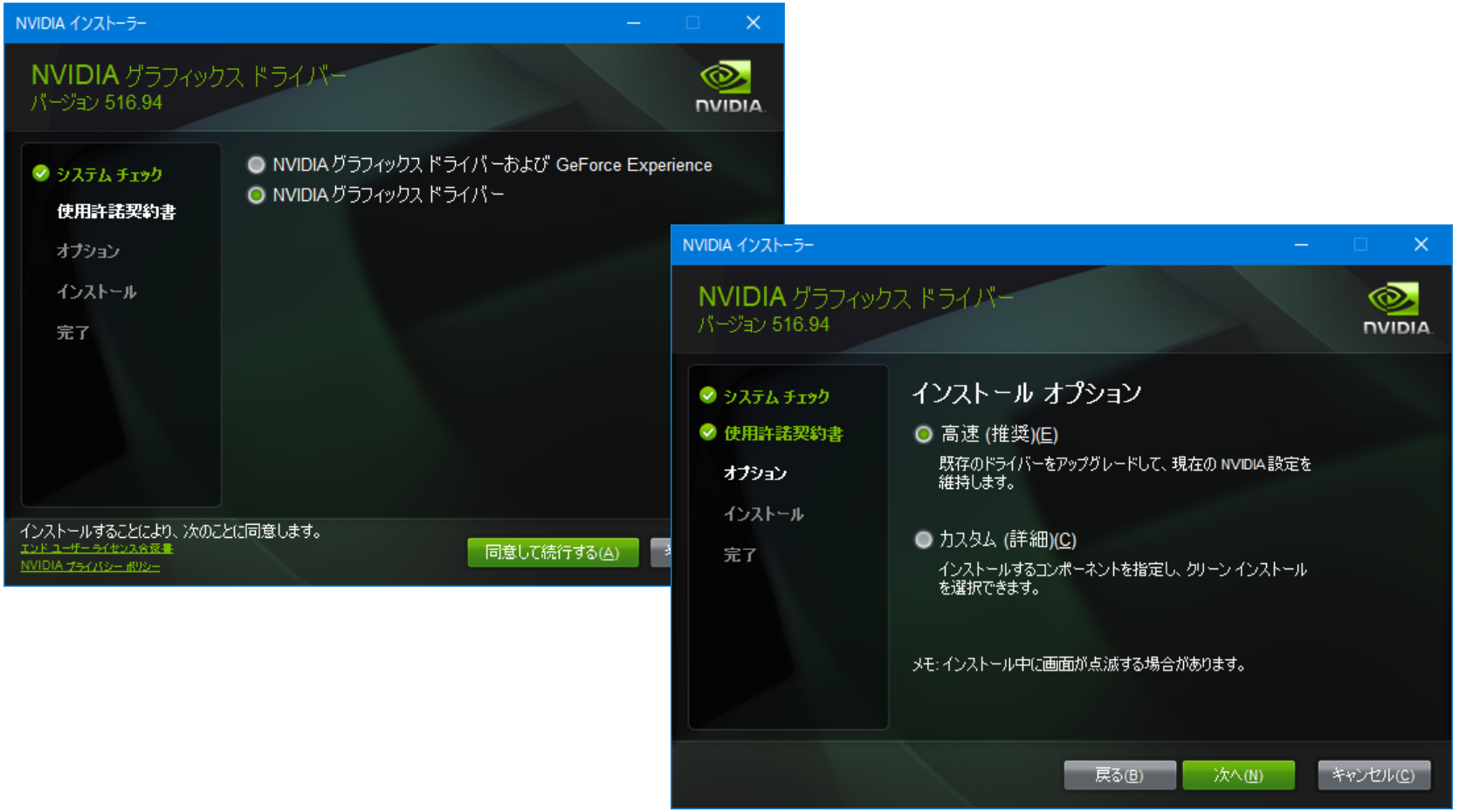
Updated Jan. 29, 2025

By H.Nishiyama, Niigata-Univ-Dent.

GPU有

GPUのドライバを最新版にしておく

- https://www.nvidia.com/ja-jp/drivers/



GPU有

搭載されているGPUに応じて制約を確認する

- 搭載されているGPUが利用可能か確認する（NVIDIAの場合）
  - https://developer.nvidia.com/cuda-gpus

- ドライバのバージョンにて利用可能なCUDAのバージョンをチェックする。
  - https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html

GPU有

Table 2: CUDA Toolkit and Minimum Required Driver Version for CUDA Minor Version Compatibility

CUDA Toolkit	Minimum Required Driver Version for CUDA Minor Version Compatibility*	
	Linux x86_64 Driver Version	Windows x86_64 Driver Version
CUDA 12.x	>=525.60.13	>=528.33
CUDA 11.8.x CUDA 11.7.x CUDA 11.6.x CUDA 11.5.x CUDA 11.4.x CUDA 11.3.x CUDA 11.2.x CUDA 11.1.x	>=450.80.02	>=452.39
CUDA 11.0 (11.0.3)	>=450.36.06**	>=451.22**

* Using a Minimum Required Version that is **different** from Toolkit Driver Version could be allowed in compatibility mode – please read the CUDA Compatibility Guide for details.

** CUDA 11.0 was released with an earlier driver version, but by upgrading to Tesla Recommended Drivers 450.80.02 (Linux) / 452.39 (Windows), minor version compatibility is possible across the CUDA 11.x family of toolkits.

NVIDIAのドライババージョンから、利用可能なCUDAを確認する。
※予めNVIDIAのドライバは最新のものに更新しておくこと。

Table 3: CUDA Toolkit and Corresponding Driver Versions

CUDA Toolkit	Toolkit Driver Version	
	Linux x86_64 Driver Version	Windows x86_64 Driver Version
CUDA 12.8 GA	>=570.26	>=570.65
CUDA 12.6 Update 3	>=560.35.05	>=561.17
CUDA 12.6 Update 2	>=560.35.03	>=560.94
CUDA 12.6 Update 1	>=560.35.03	>=560.94
CUDA 12.6 GA	>=560.28.03	>=560.76
CUDA 12.5 Update 1	>=555.42.06	>=555.85
CUDA 12.5 GA	>=555.42.02	>=555.85
CUDA 12.4 Update 1	>=550.54.15	>=551.78

TensorFlowかPyTorchか？

必要なPythonやCUDAのバージョンが異なるので注意

- PyTorch
  - 数年前から利用率がTensorFlowを上回り、テキストおよびネット上のサンプルも増えてきている。
  - APIが低レベルのため、プログラミングの難易度が高くなる傾向あり、pipenv環境へのインストールも難しくなっていたが、ここ1，2年で劇的に使い勝手が良くなった。
  - CUDAおよびcuDNNが一体化しており、OSレベルでインストールされたバージョンとは無関係なので、pipenv環境などでも使いやすい。
- TensorFlow
  - 現行利用している教科書と演習課題の一部にて使用している。
  - TensorFlow1の時代と比較し2の時代のものは内部構造が理解しにくくなっている。
  - CUDAおよびcuDNNがシステムにインストールされていないと使えないため、Docker使うか、システム（OS）レベルでの切り換えが必要
  - Tensorflowを含めライブラリのバージョンが上がっていくと、古いソフトが動かないことが多いので、Docker等の利用を考えた方が良いと思われる。

GPU有

NVIDIA-GPUをpipenv systemで使う

- 適するcuDNN, Python および ソフトの組み合わせバージョンを確認（重要）
  - PyTorchの場合は https://pytorch.org/get-started/locally/
  - TensorFlowは https://www.tensorflow.org/install/source_windows?hl=ja#gpu
- TensorFlowでは適切なCUDA toolkit と cuDNN を別にダウンロードして、インストールする

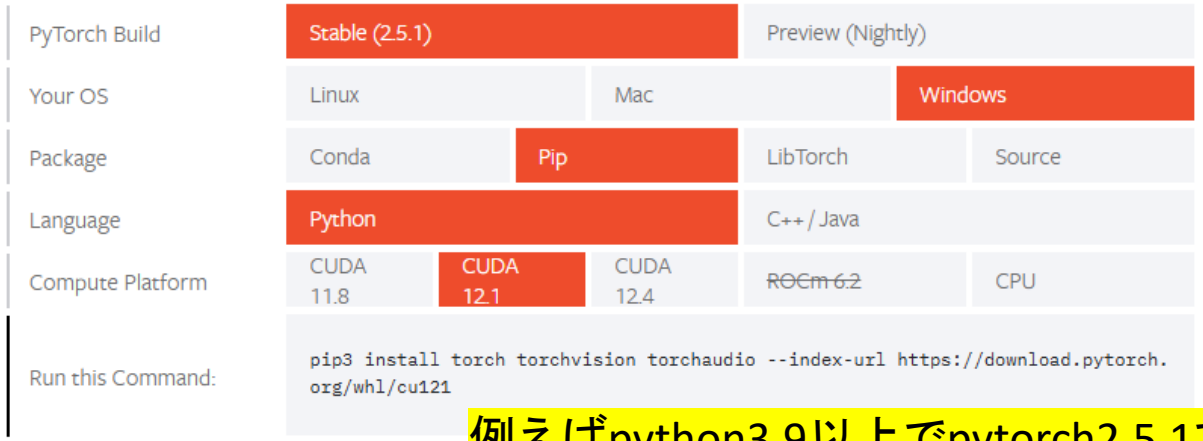
GPU有

PyTorchの場合

Pipenv利用時、まずはPipとして適合システム確認※

- PyTorchのサイト：https://pytorch.org/
  - 上記内から下記のインストールに入り、自身のシステムを確認
  - https://pytorch.org/get-started/locally/
  - CUDAを含んだライブラリとしてインストールされる

NOTE: Latest PyTorch requires Python 3.9 or later.



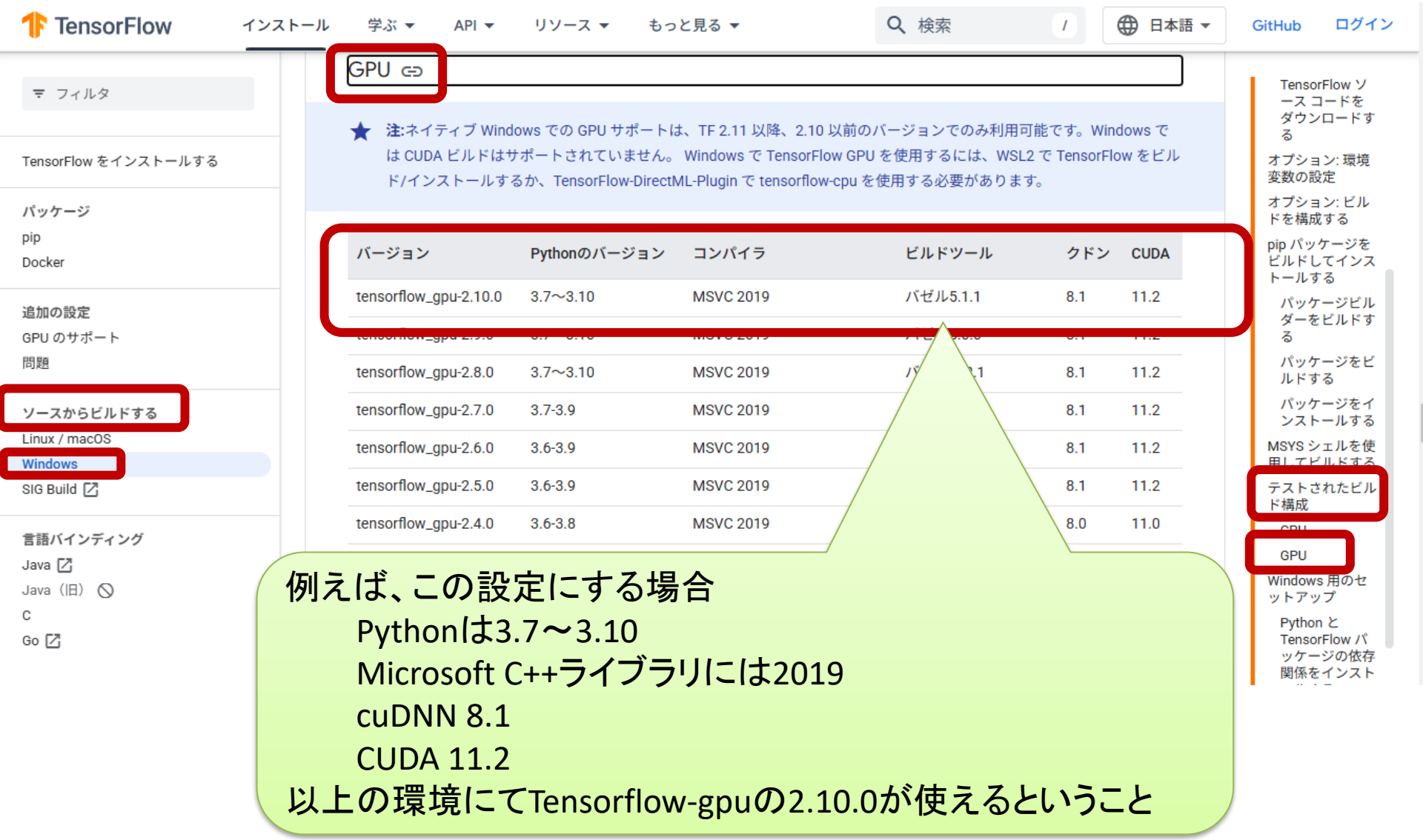
例えばpython3.9以上でpytorch2.5.1でwindowsでCUDA 12.1

※Pipenvの場合、Pipでのインストールとは異なるので注意。下記参照
https://qiita.com/aujinen/items/db919ceee6da8db1155c
https://www5.dent.niigata-u.ac.jp/~nisiyama/grad/Pipenv-PyTorch.pdf

GPU有

TensorFlowの場合（ここから以降）

https://www.tensorflow.org/install/source_windows?hl=ja#gpu



例えば、この設定にする場合

Pythonは3.7〜3.10

Microsoft C++ライブラリには2019

cuDNN 8.1

CUDA 11.2

以上の環境にてTensorflow-gpuの2.10.0が使えるということ

GPU有

TensorFlowの場合

Python, CUDA, cuDNNのバージョン

2024/03/02現在・稼働確認済みのもの

Python	CUDA toolkit	cuDNN	TensorFlow-GPU
3.5-3.7	10.1	7.65	1.14.0
3.6-3.9	11.2以上	8.1以上	2.6.0
3.7-3.10	11.2以上	8.1以上	2.8.0以上(※)

※Python3.10.7, CUDA11.7.1, cuDNN8.6.0, TensorFlow-GPU2.10にて確認済み

★ **Note:** GPU support on native-Windows is only available for 2.10 or earlier versions, starting in TF 2.11, CUDA build is not supported for Windows. For using TensorFlow GPU on Windows, you will need to build/install TensorFlow in WSL2 or use tensorflow-cpu with TensorFlow-DirectML-Plugin

注:ネイティブ Windows での GPU サポートは、2.10 以前のバージョンでのみ利用可能です。TF 2.11 以降、Windows では CUDA ビルドはサポートされていません。Windows で TensorFlow GPU を使用するには、WSL2 で TensorFlow をビルド/インストールするか、TensorFlow-DirectML-Plugin で tensorflow-cpu を使用する必要があります。

GPU有

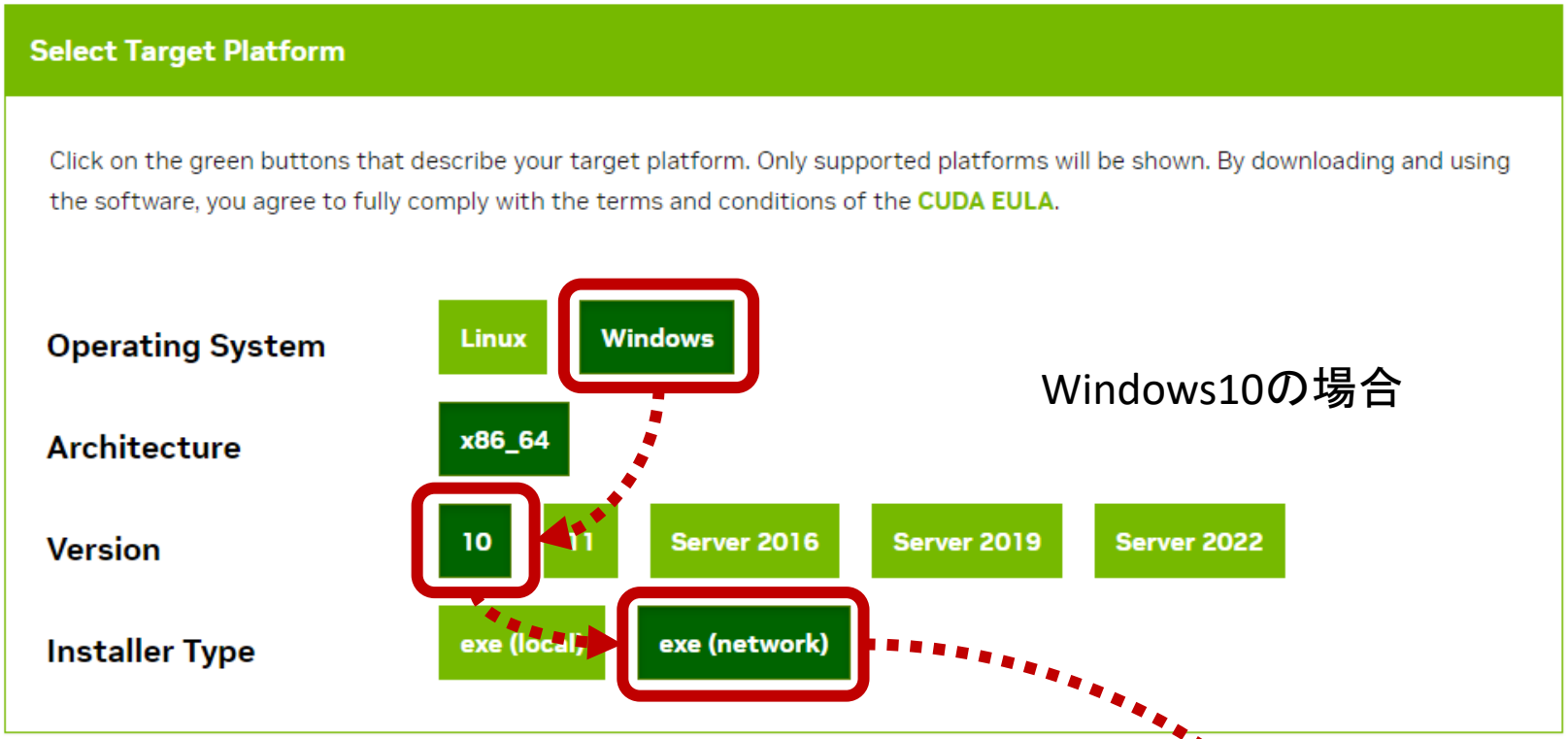
TensorFlowの場合

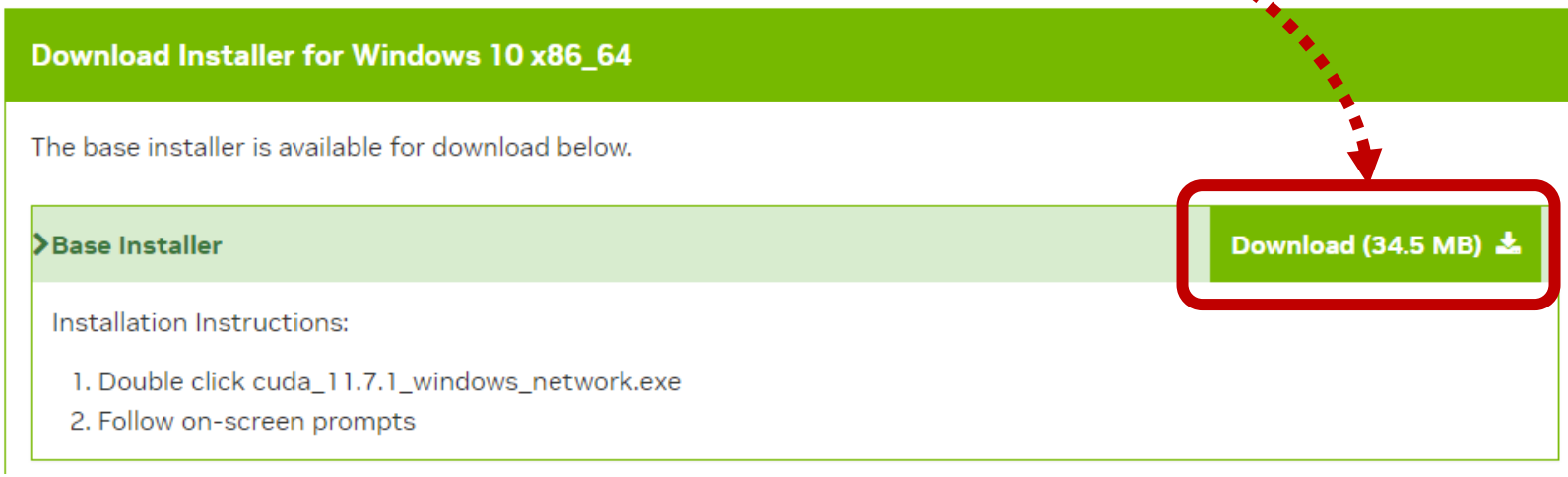
CUDAをcuda-toolkit-archiveからインストールする

- https://developer.nvidia.com/cuda-toolkit-archive

TensorFlowの場合

GPU有

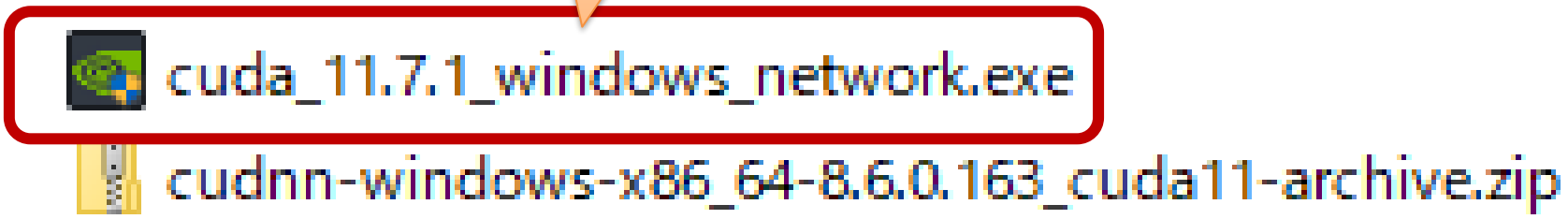




TensorFlowの場合

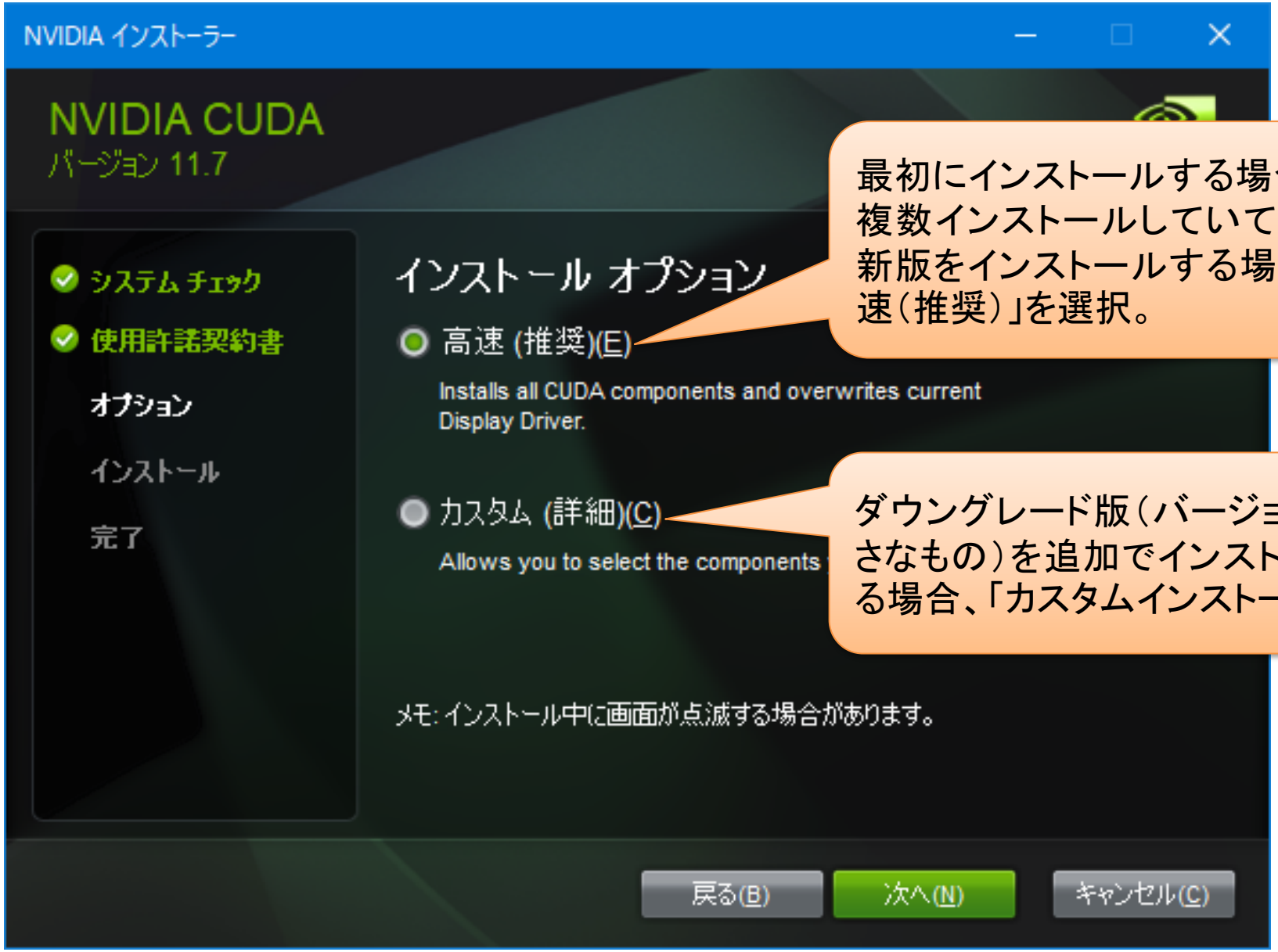
GPU有

ダウンロードしたインストーラーを
ダブルクリックしてインストール



TensorFlowの場合

GPU有

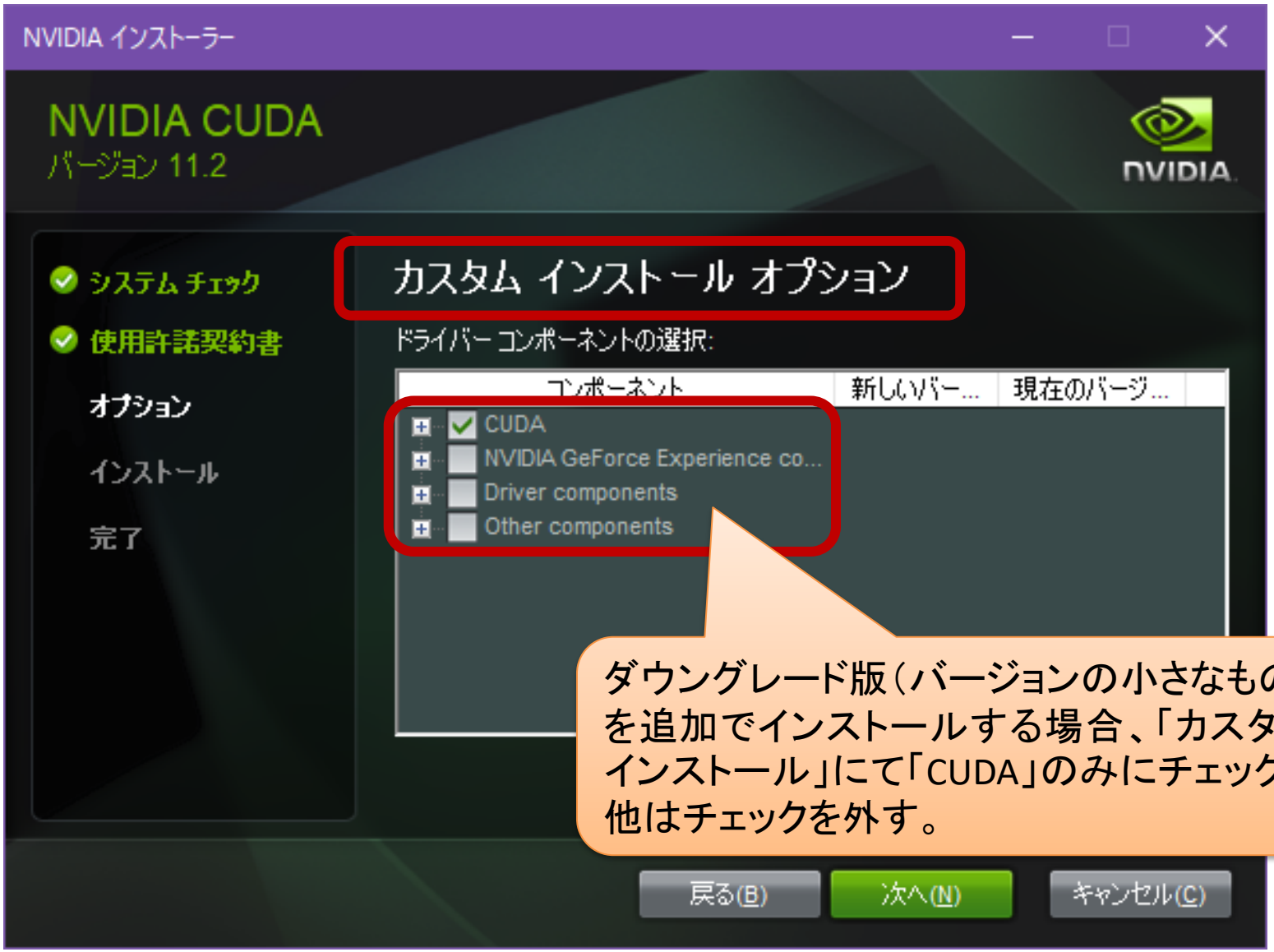


最初にインストールする場合や、複数インストールしていても、最新版をインストールする場合「高速（推奨）」を選択。

ダウングレード版（バージョンの小さなもの）を追加でインストールする場合、「カスタムインストール」

GPU有

カスタムインストールの場合



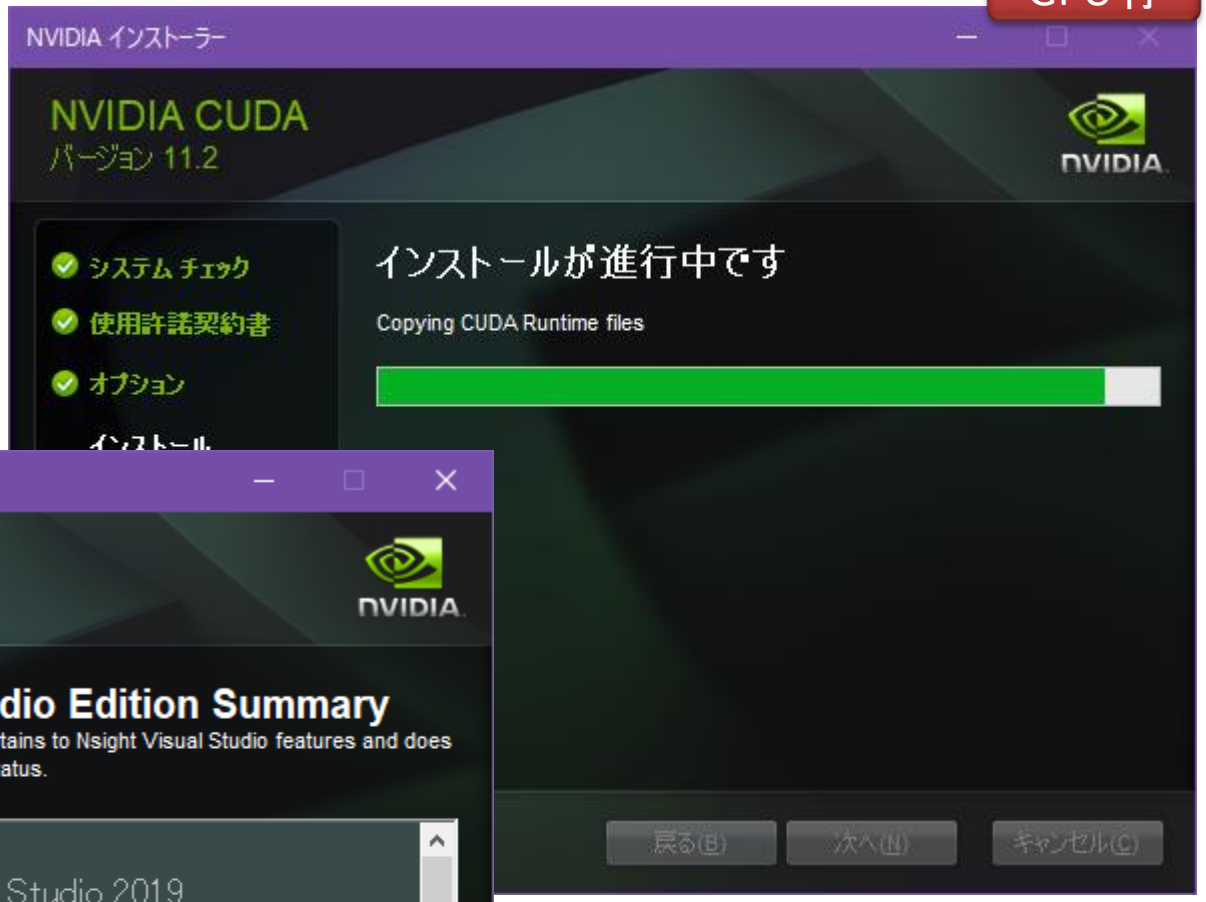
TensorFlowの場合

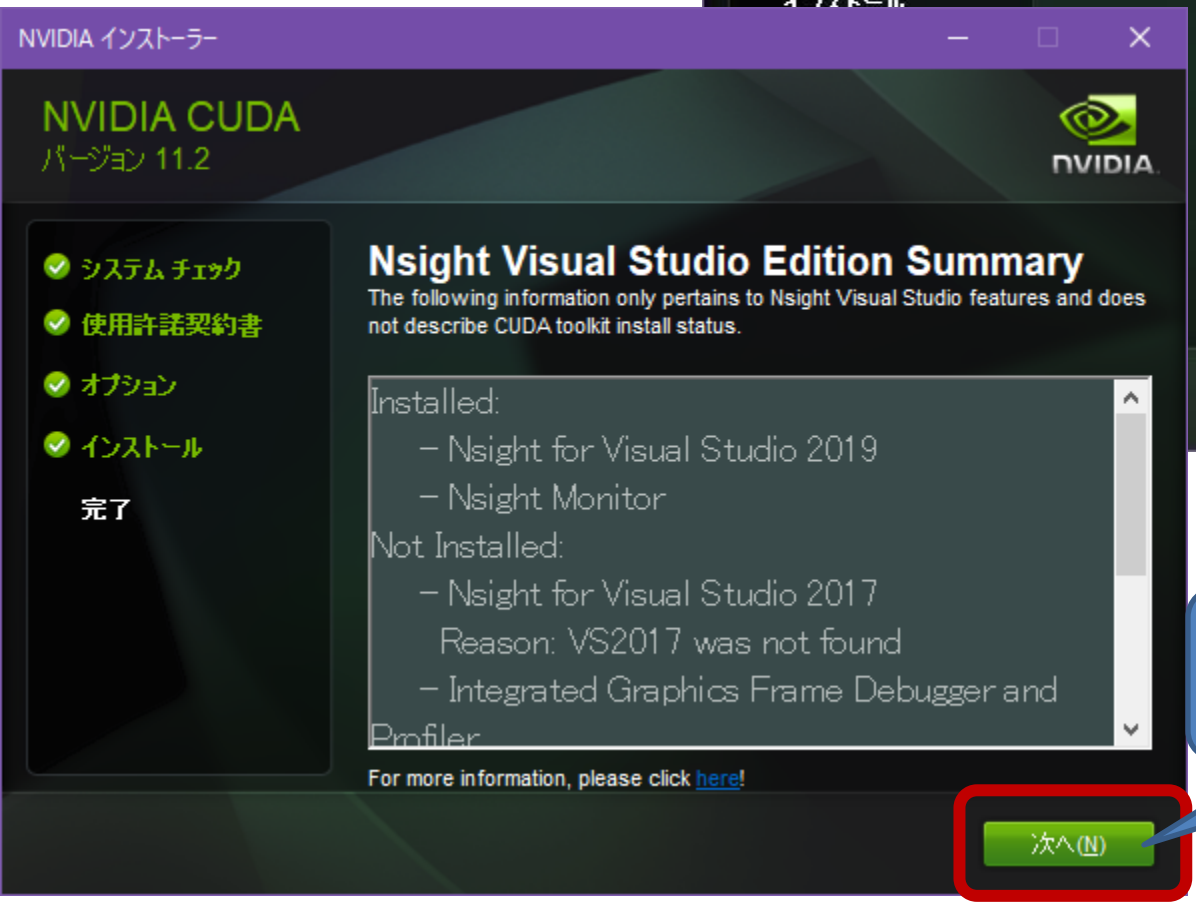
GPU有

【次へ】をクリック・・・

TensorFlowの場合

GPU有





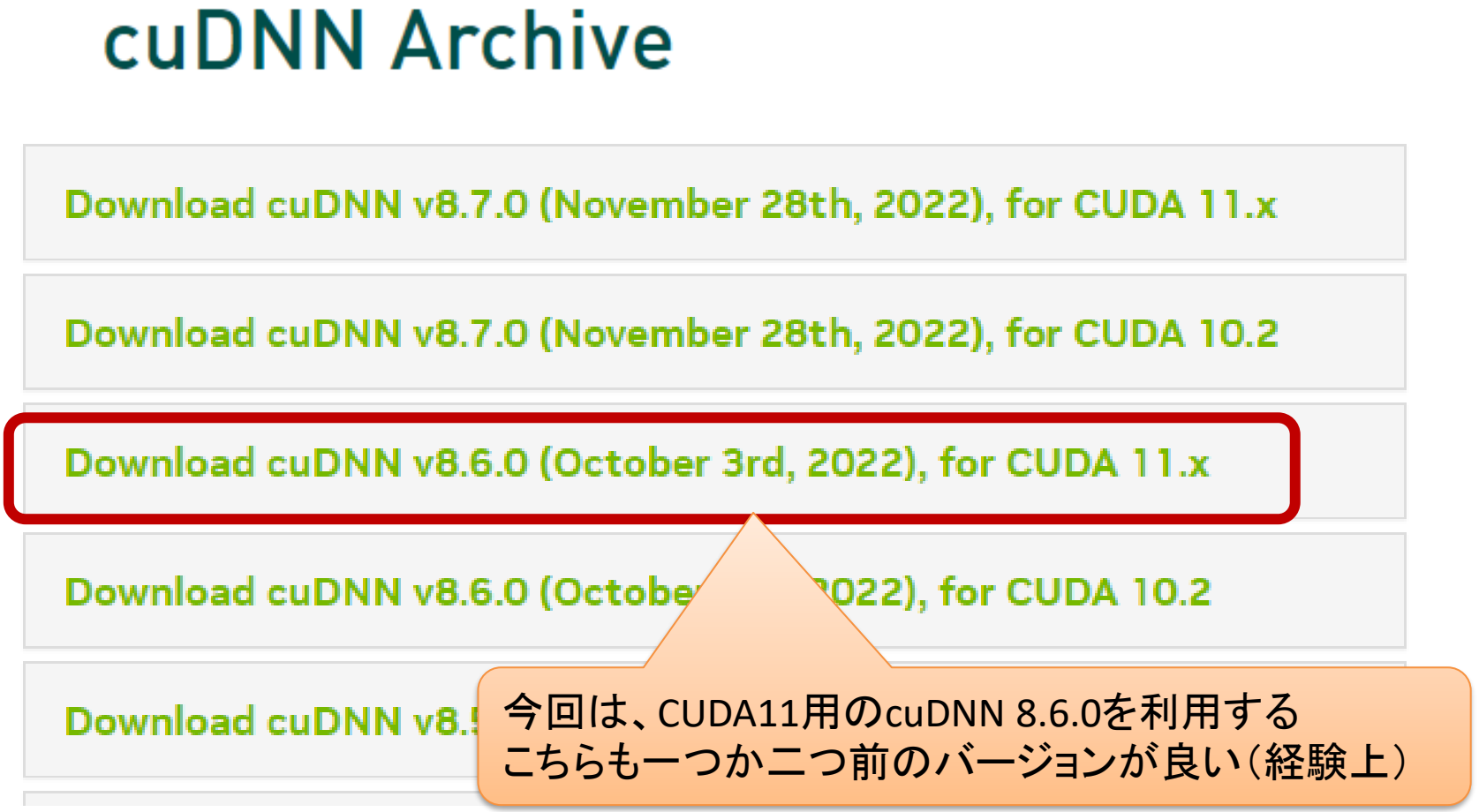
【次へ】をクリックし、終了

TensorFlowの場合

GPU有

cuDNNをcuDNN Archiveからダウンロードする

- https://developer.nvidia.com/rdp/cudnn-archive



TensorFlowの場合

GPU有

Download cuDNN v8.7.0 (November 28th, 2022), for CUDA 10.2

Download cuDNN v8.6.0 (October 3rd, 2022), for CUDA 11.x

クリックすると下記が展開・表示される

Local Installers for Windows and Linux, Ubuntu(x86_64, armsbsa)

Local Installer for Windows (Zip)

Windows用を選択する

Local Installer for Linux x86_64 (Tar)

Local Installer for Linux PPC (Tar)

Local Installer for Linux SBSA (Tar)

Local Installer for Ubuntu18.04 x86_64 (Deb)

Local Installer for Debian 11 (Deb)

Local Installer for Ubuntu20.04 x86_64 (Deb)

Local Installer for Ubuntu22.04 x86_64 (Deb)

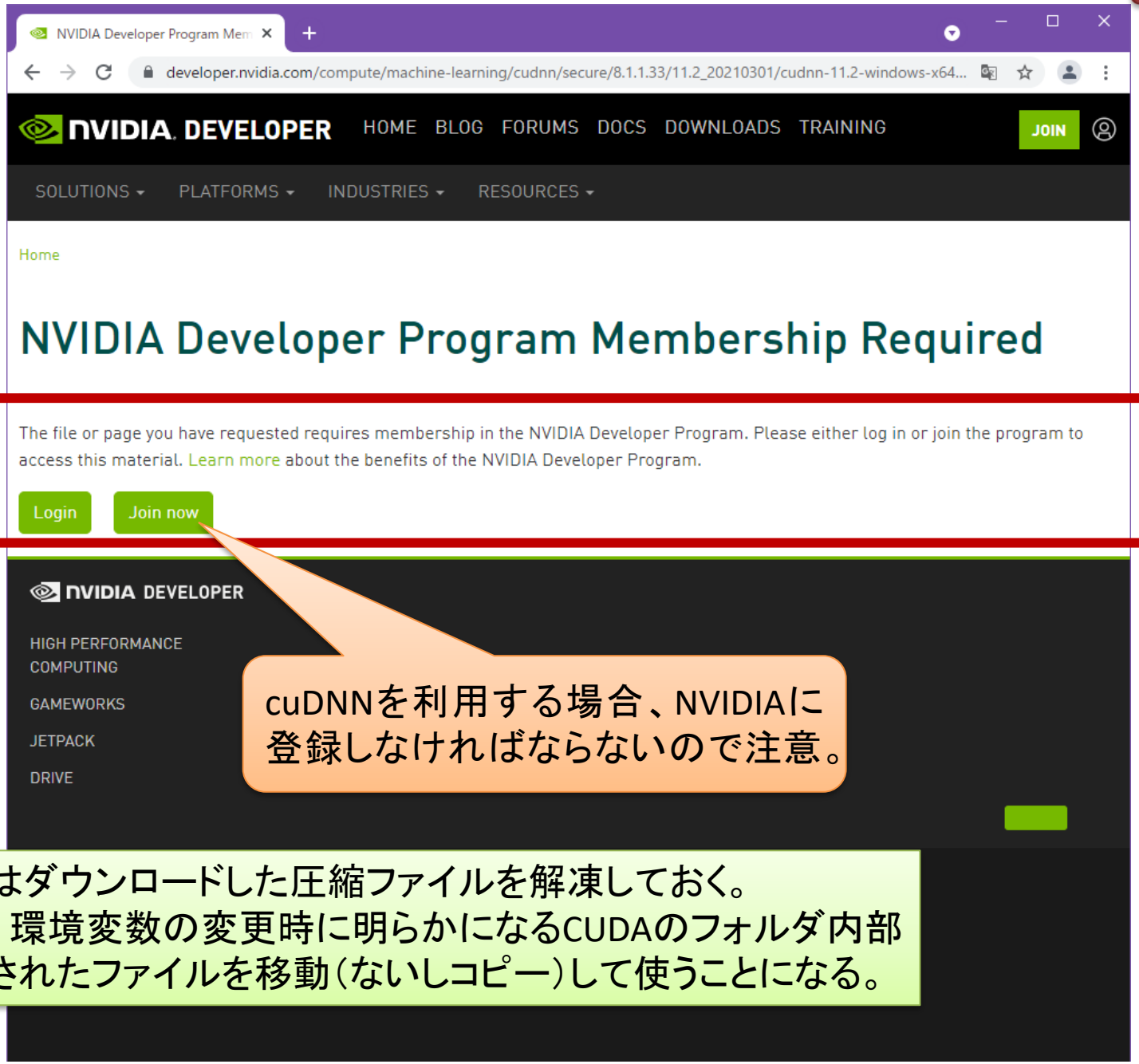
Local Installer for Ubuntu20.04 aarch64sbsa (Deb)

Local Installer for Ubuntu20.04 cross-sbsa (Deb)

Local Installers for Red Hat (x86_64,

TensorFlowの場合

GPU有

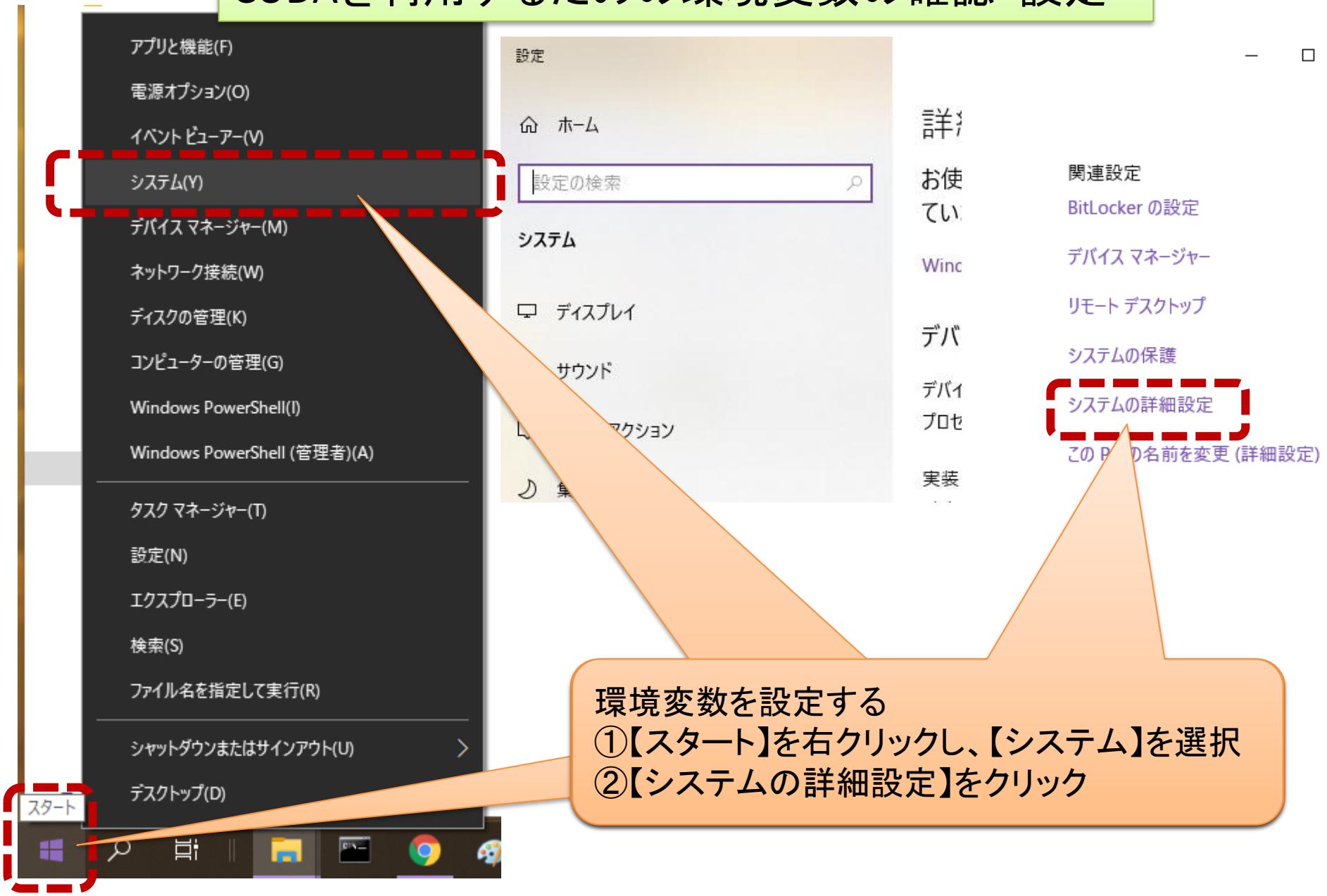


cuDNNはダウンロードした圧縮ファイルを解凍しておく。
この後、環境変数の変更時に明らかになるCUDAのフォルダ内部に解凍されたファイルを移動（ないしコピー）して使うことになる。

TensorFlowの場合

GPU有

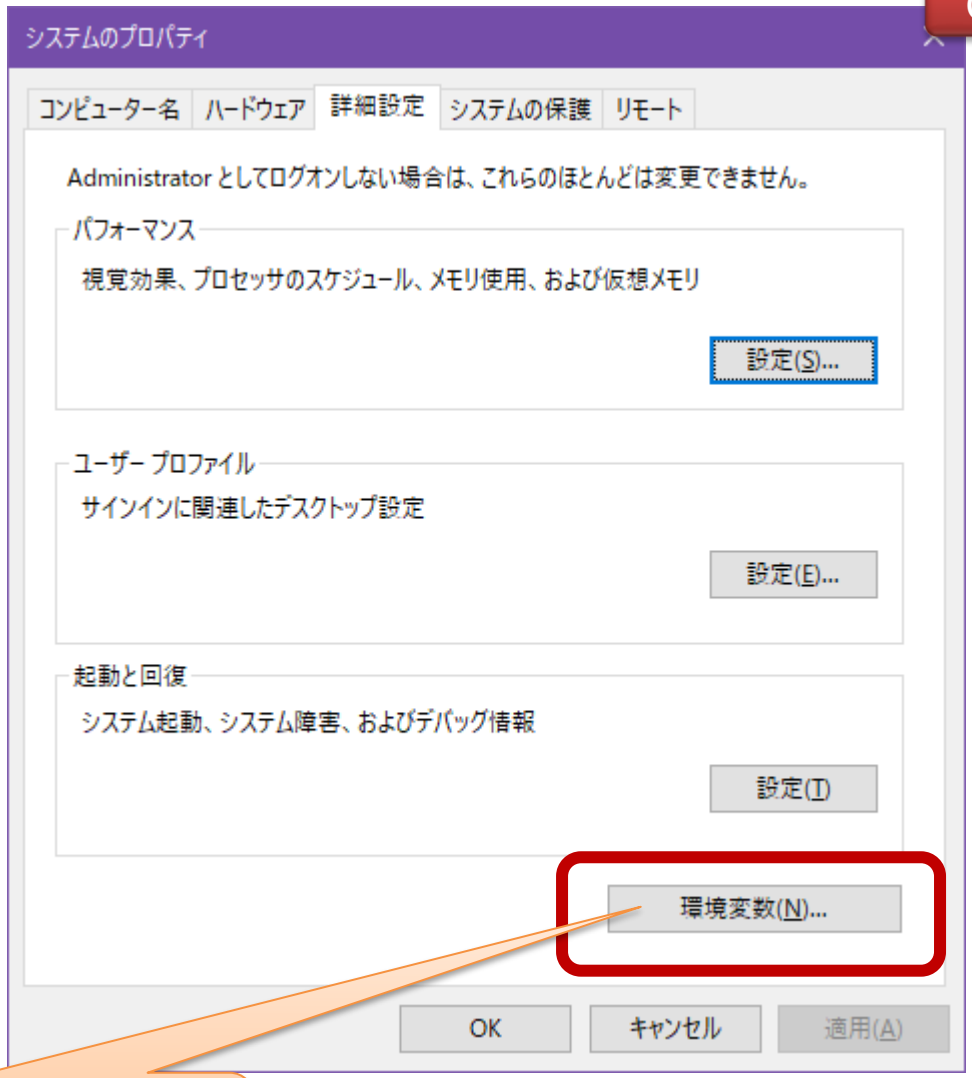
CUDAを利用するための環境変数の確認・設定



環境変数を設定する
①【スタート】を右クリックし、【システム】を選択
②【システムの詳細設定】をクリック

TensorFlowの場合

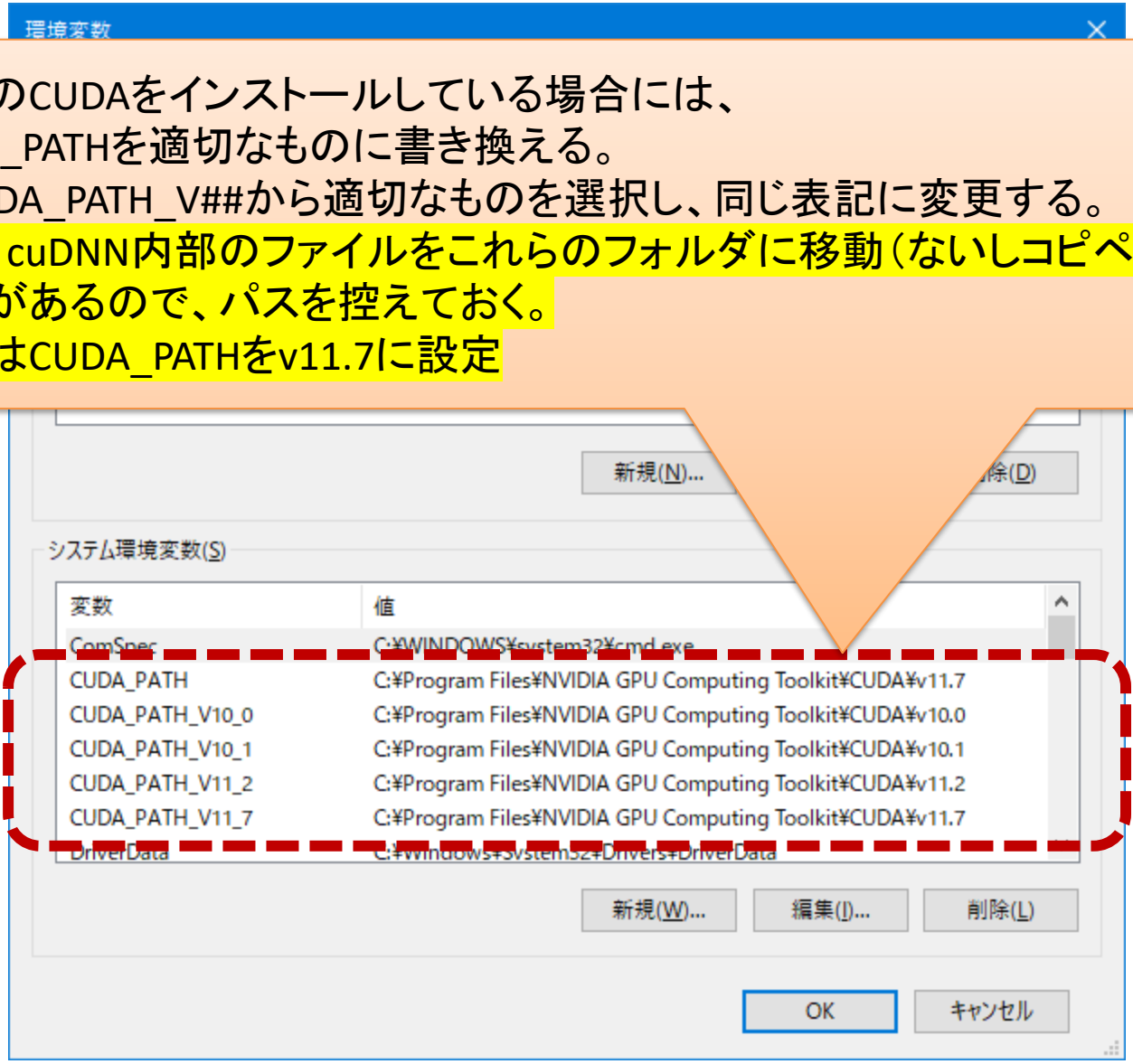
GPU有



環境変数... をクリック

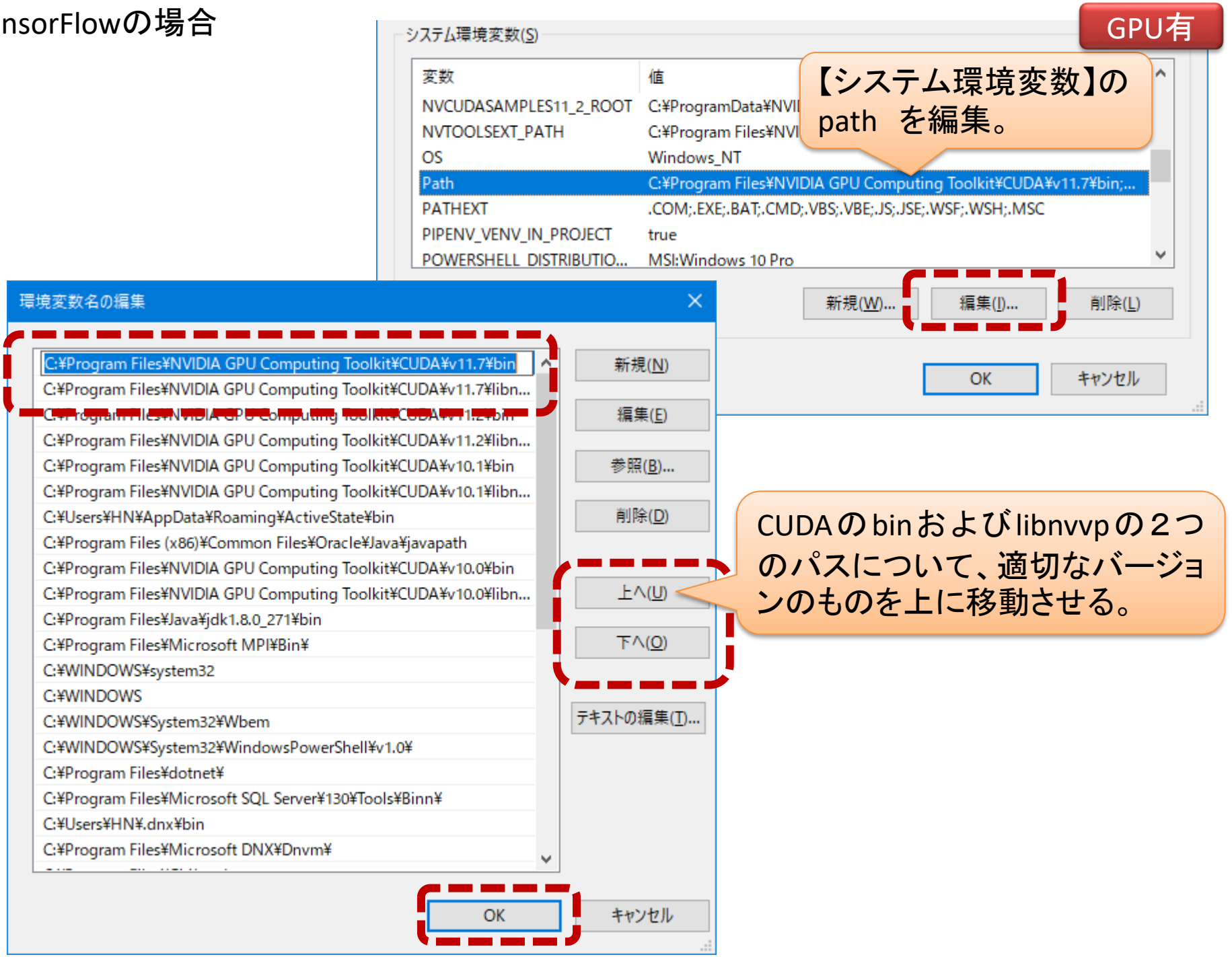
TensorFlowの場合

GPU有



複数のCUDAをインストールしている場合には、
CUDA_PATHを適切なものに書き換える。
※CUDA_PATH_V##から適切なものを選択し、同じ表記に変更する。
また、cuDNN内部のファイルをこれらのフォルダに移動（ないしコピペ）する必要があるので、パスを控えておく。
今回はCUDA_PATHをv11.7に設定

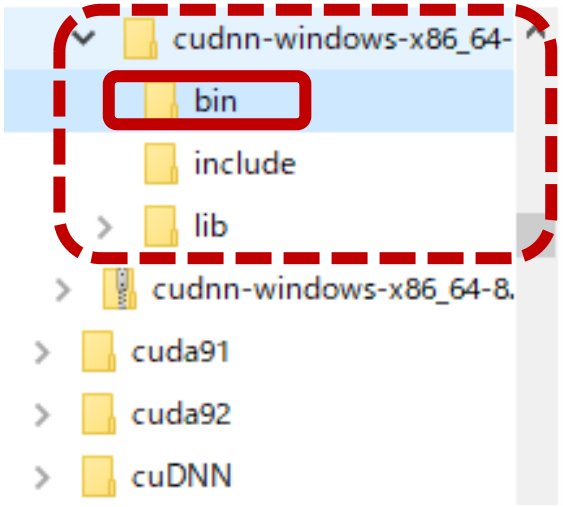
TensorFlowの場合

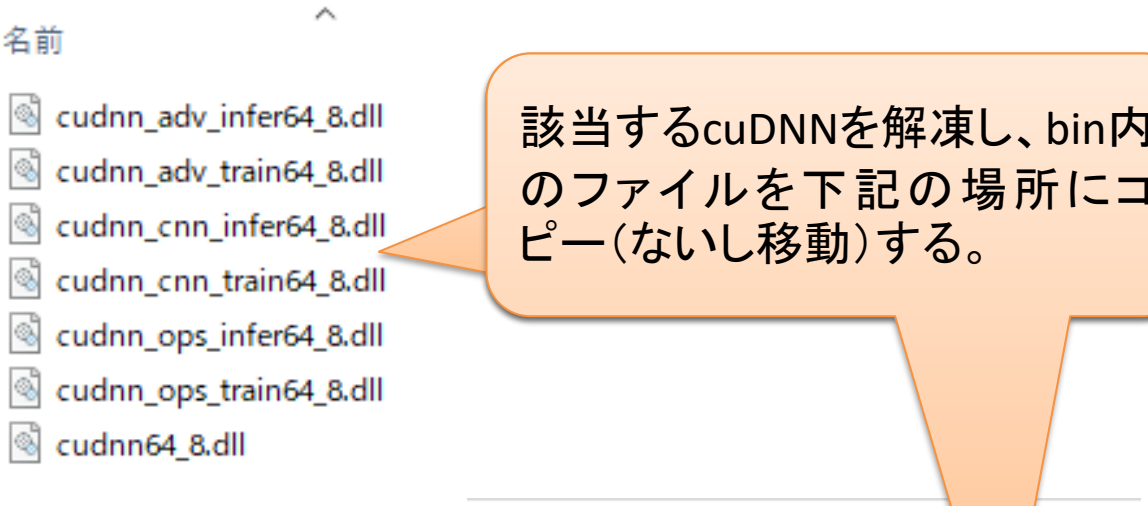


TensorFlowの場合

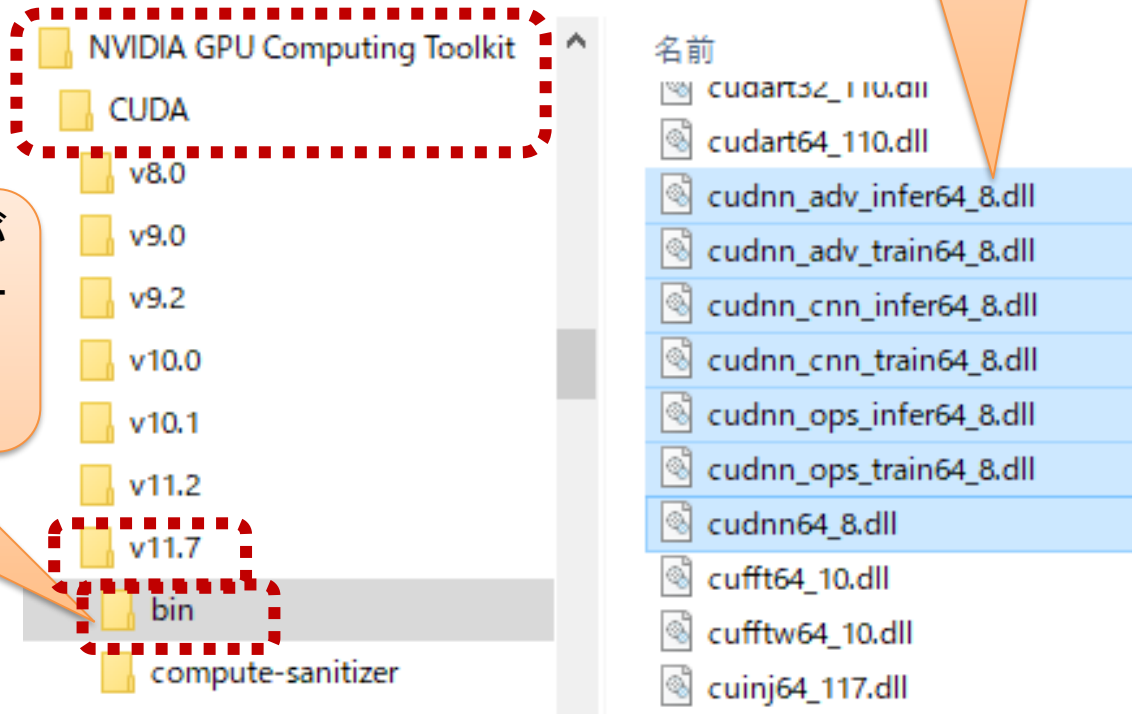
GPU有

cuDNNからCUDAへのファイルの移動（ないしコピー）





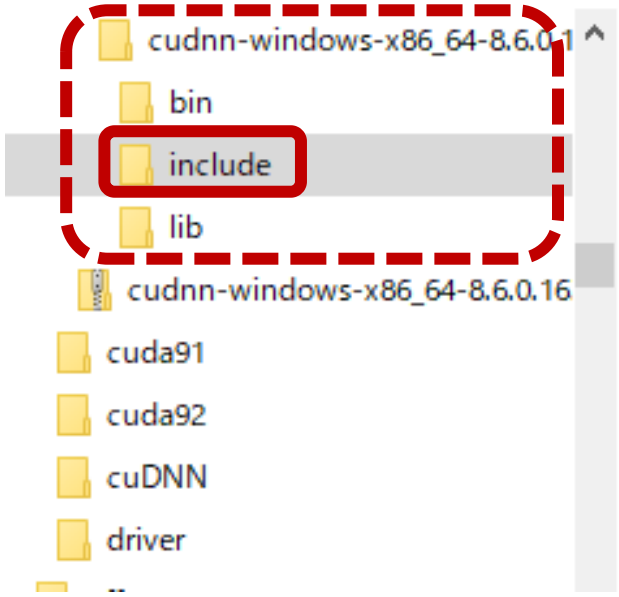
該当するcuDNNを解凍し、bin内のファイルを下記の場所にコピー（ないし移動）する。

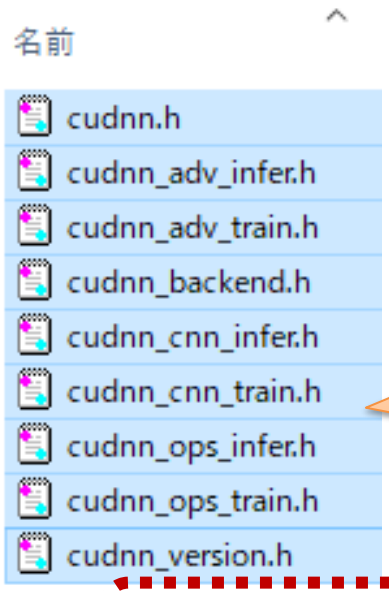


該当するバージョンのCUDAがインストールされた場所（システム環境変数にて控えた場所）内部のbinフォルダ

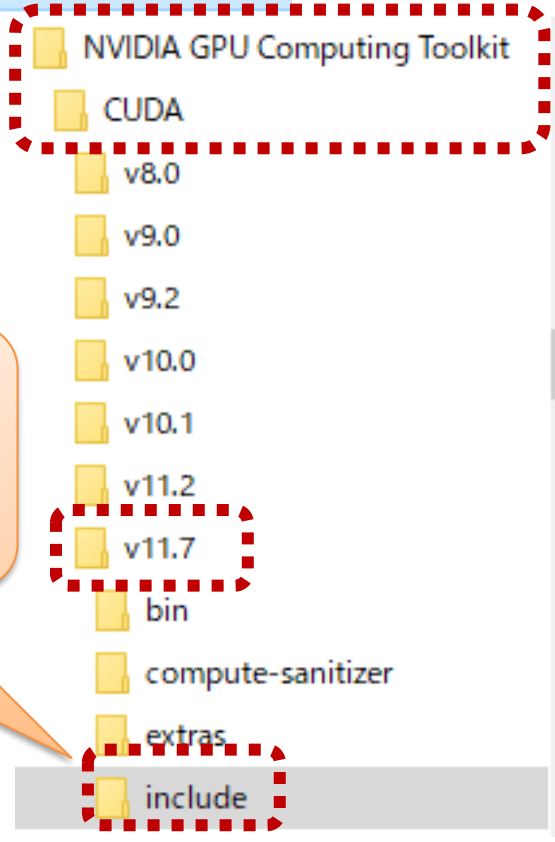
TensorFlowの場合

GPU有

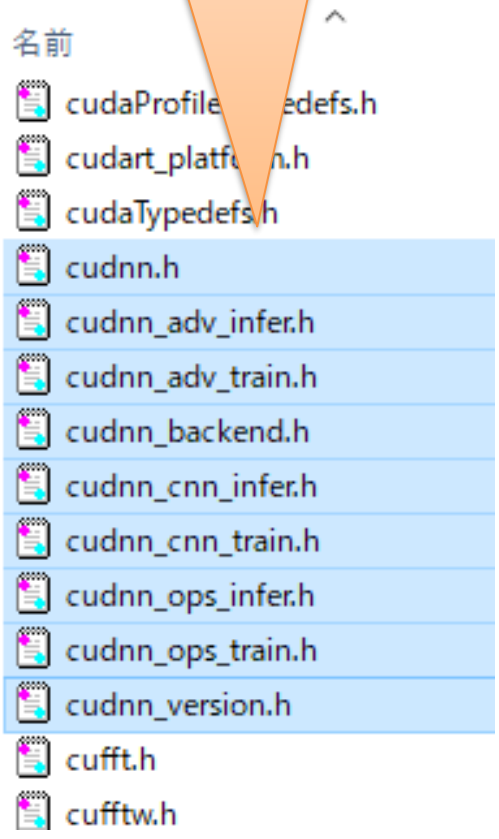




解凍したcuDNNの、 include内のファイルを下記の場所にコピー(ないし移動)する。

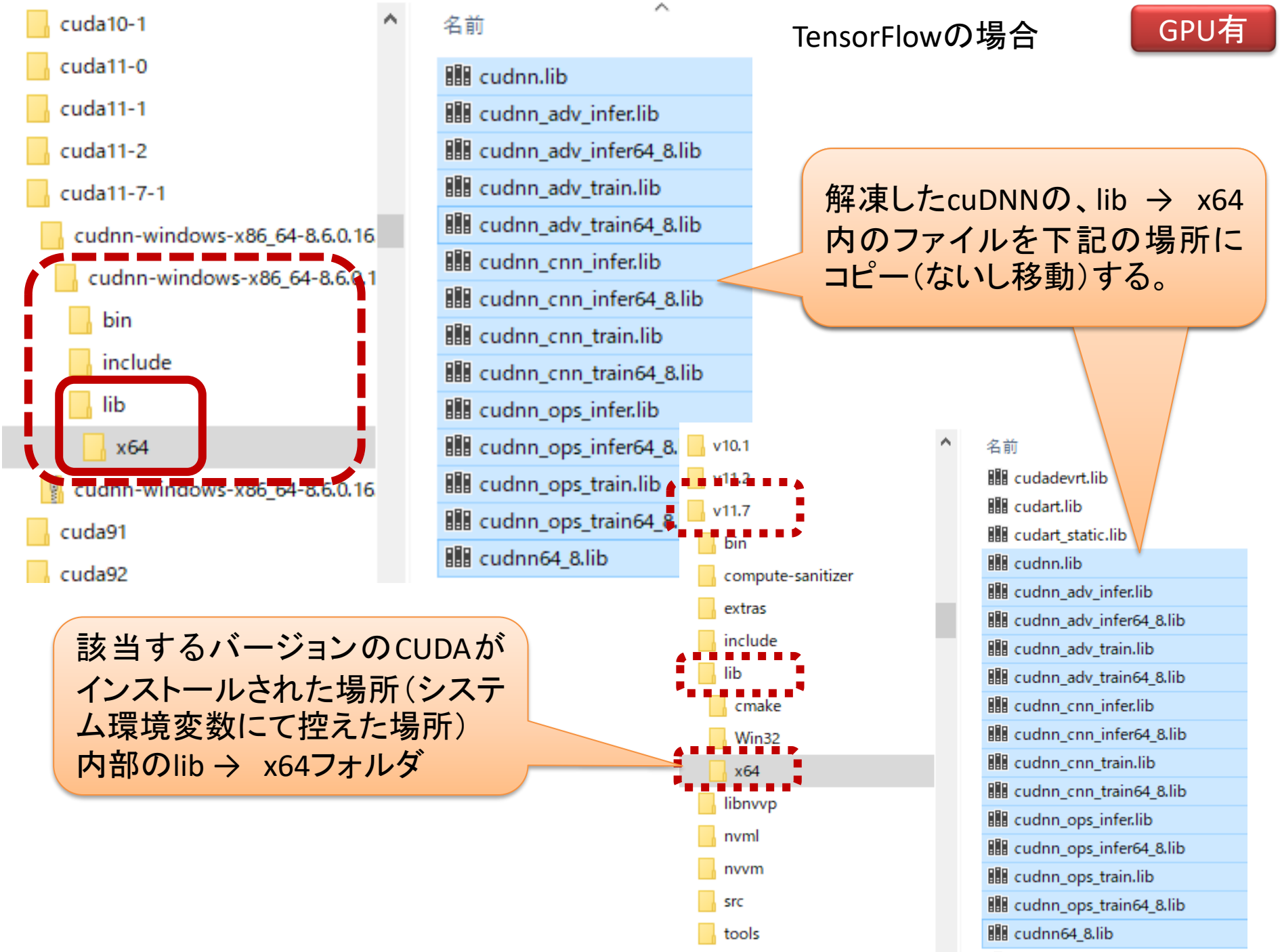


該当するバージョンのCUDAがインストールされた場所(システム環境変数にて控えた場所)内部のincludeフォルダ



TensorFlowの場合

GPU有



TensorFlowの場合

CUDAおよびcuDNNのバージョン確認

- CUDAのバージョン確認
  - nvcc –V

通常releaseの数字の前に「v」を付けたフォルダ内に下記cuDNN関連のファイルも入っている。

```
nvcc: NVIDIA (R) Cuda compiler dri
Copyright (c) 2005-2022 NVIDIA Cor ration
Built on Wed_Jun__8_16:59:34_Pacific_Daylight_Time_2022
Cuda compilation tools, release 11.7, V11.7.99
Build cuda_11.7.r11.7/compiler.31442593_0
```

- cuDNNのバージョン確認
  - type "C:¥Program Files¥NVIDIA GPU Computing Toolkit¥CUDA¥v***¥include¥cudnn_version.h“
  - 「 ***」の部分を上記のCUDAのバージョンに変え「#define」の部分にて確認。

```
#define CUDNN_MAJOR 8
#define CUDNN_MINOR 6
#define CUDNN_PATCHLEVEL 0
```

GPU有

NVIDIA-GPUをpipenv systemで使う (1).

- “Visual Studio 20xx の Microsoft Visual C++ 再頒布可能パッケージ” をダウンロードし、インストールする。
  - https://visualstudio.microsoft.com/ja/vs/older-downloads/
  - “Visual Studio Dev Essentials”への入会が必要
  - 「最新の Microsoft Visual C++ 再頒布可能パッケージ バージョン」をダウンロードしてインストールする。

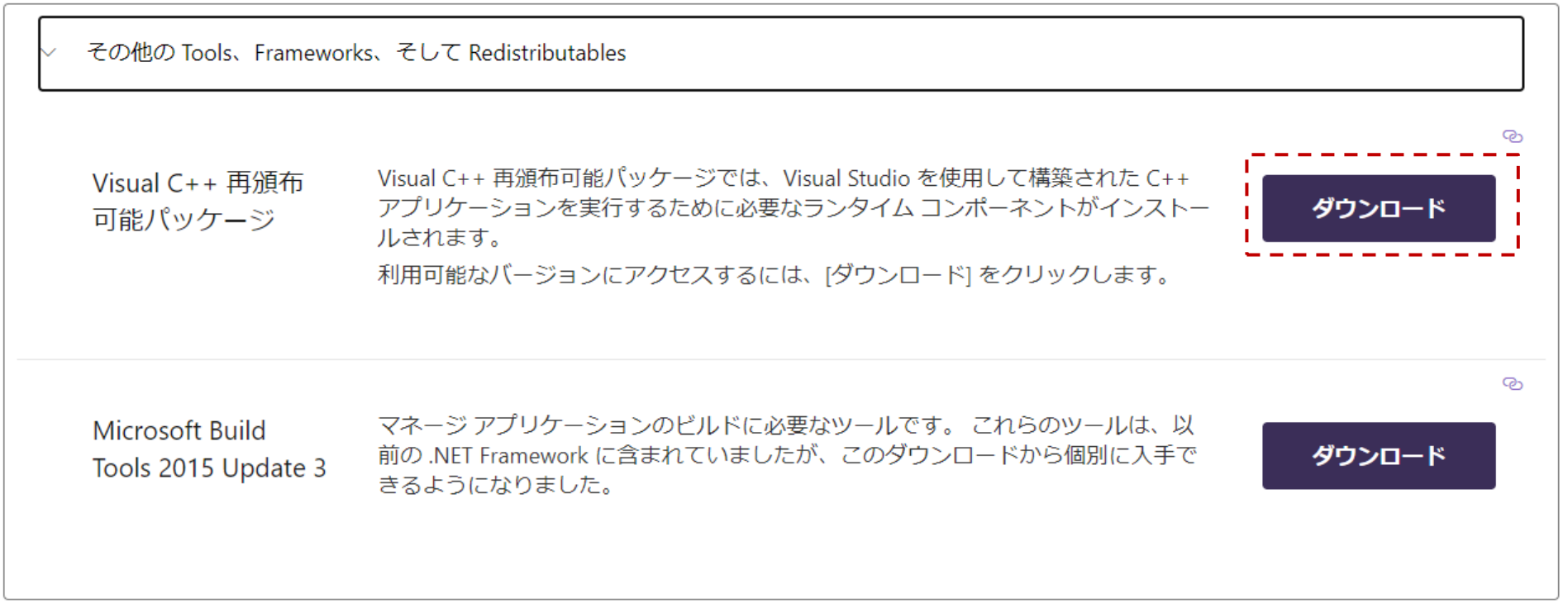
visualstudio.microsoft.com/ja/vs/older-downloads/

Visual Studio サブスクライバーのダウンロード

古いバージョンでは、アクティブな Visual Studio Subscription が必要です。 以下の製品を選択し、[ダウンロード] ボタンをクリックして Visual Studio (MSDN) サブスクリプションにログインし、古いバージョンにアクセスします。

すべて展開 → | すべて折りたたむ →

- 2019
- 2017
- 2015
- Visual Studio for Mac
- 分離シェルと統合シェル
- その他の Tools、Frameworks、そして Redistributables

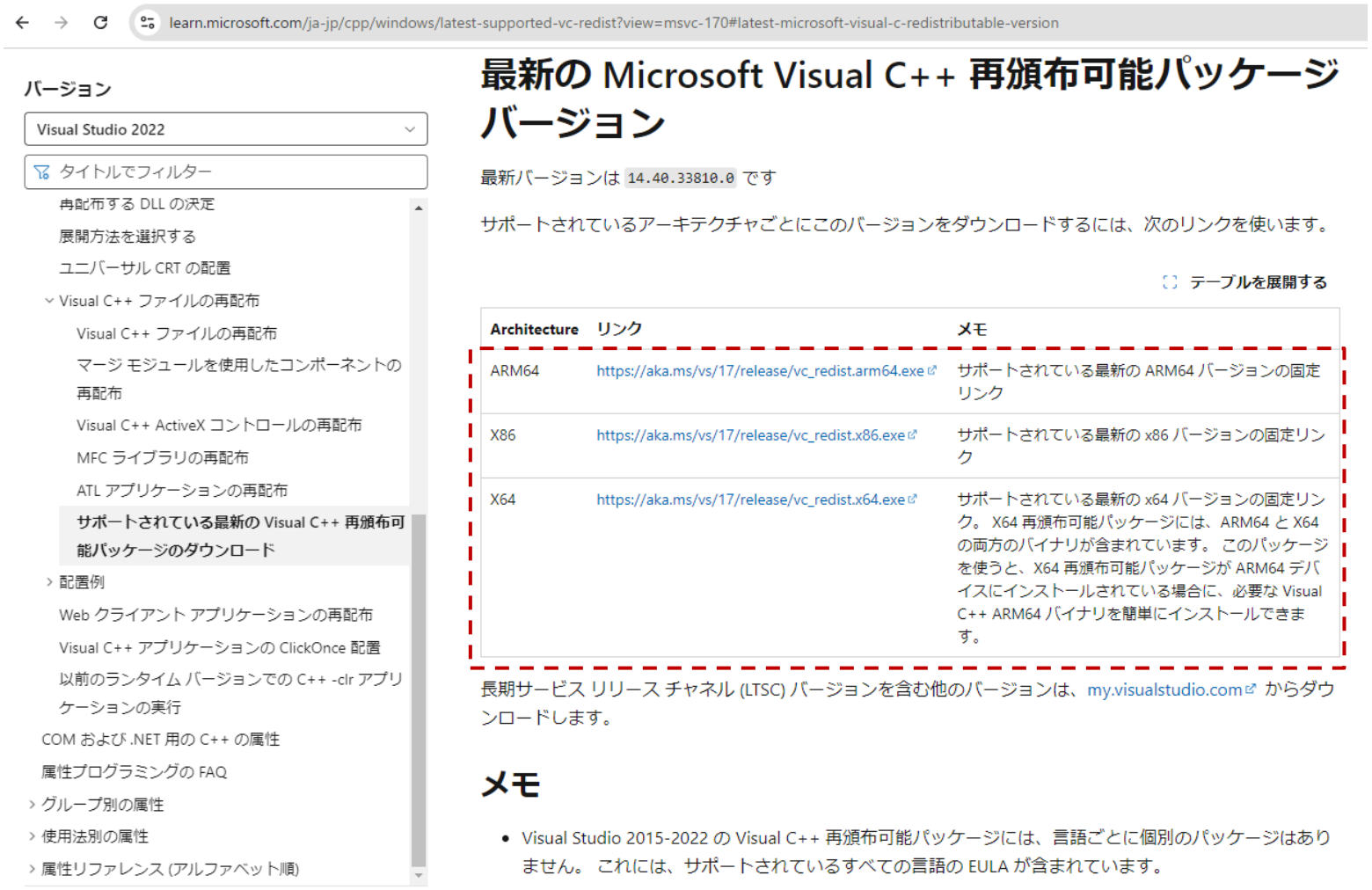

その他の Tools、Frameworks、そして Redistributables
Visual C++ 再頒布可能パッケージ
Visual C++ 再頒布可能パッケージでは、Visual Studio を使用して構築された C++ アプリケーションを実行するために必要なランタイム コンポーネントがインストールされます。
利用可能なバージョンにアクセスするには、[ダウンロード] をクリックします。
ダウンロード
Microsoft Build Tools 2015 Update 3
マネージ アプリケーションのビルドに必要なツールです。 これらのツールは、以前の .NET Framework に含まれていましたが、このダウンロードから個別に入手できるようになりました。
ダウンロード

GPU有

https://learn.microsoft.com/ja-jp/cpp/windows/latest-supported-vc-redist?view=msvc-170

GPU有

https://learn.microsoft.com/ja-jp/cpp/windows/latest-supported-vc-redist?view=msvc-170

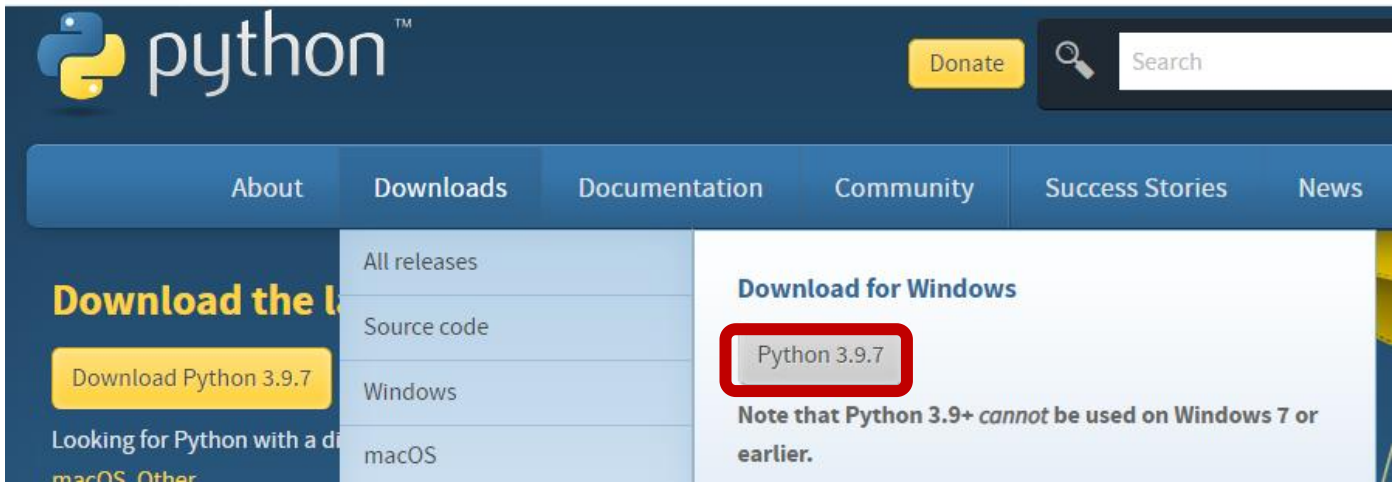


Tensorflowに伴う
GPU周りの設定はここまで

共通

Python3

- Pythonのサイト
  - https://www.python.org/
- Python3をダウンロードする
  - python2もあるが、古いコード体系なので注意



共通

インストールしようとしているアプリは、Microsoft 検証済みアプリではありません

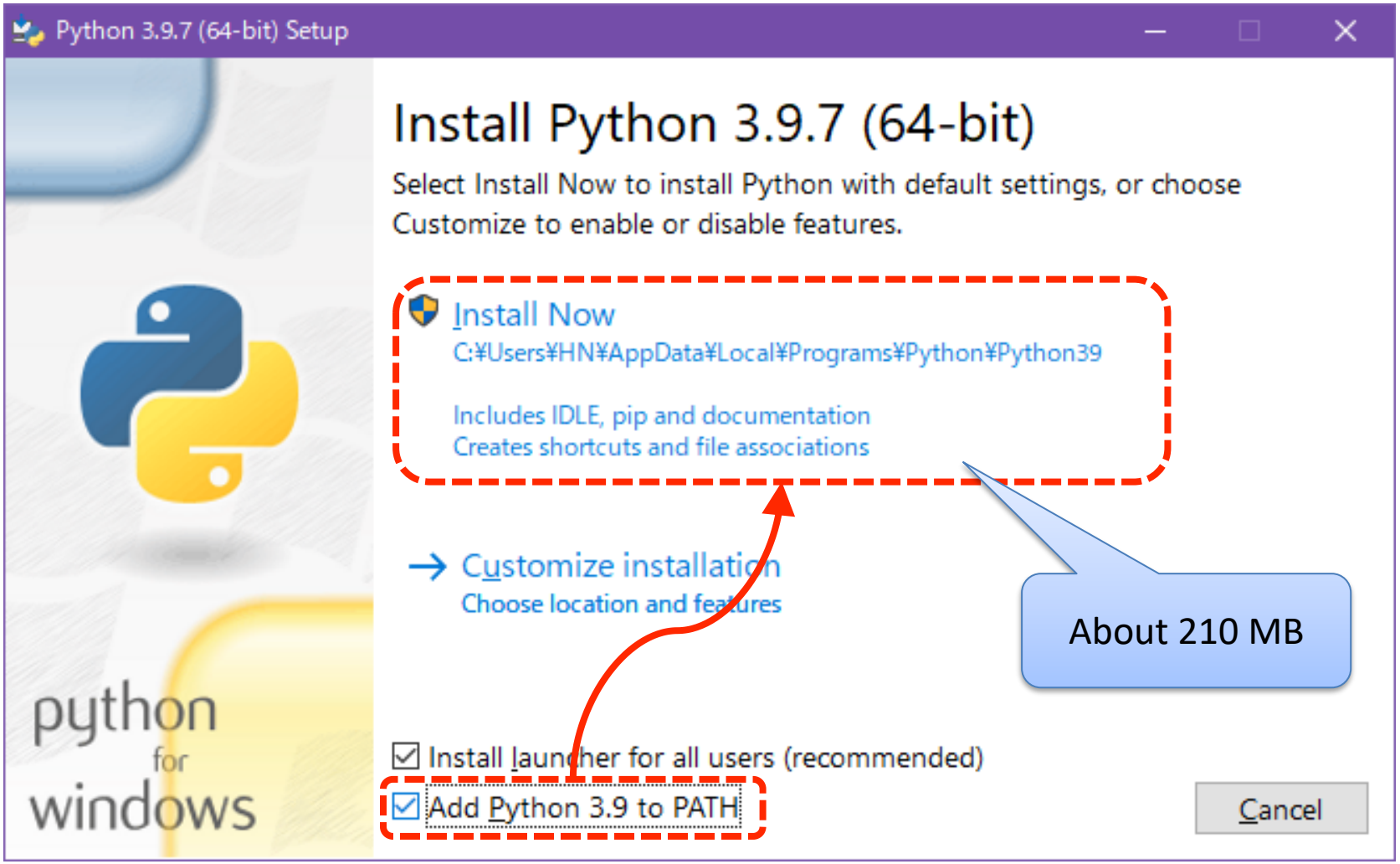
インストール対象を Microsoft Store のアプリのみに限定すると、PC の保護とスムーズな実行に役立ちます。

アプリを Microsoft Store から入手　インストールする

アプリのおすすめの設定を変更 >

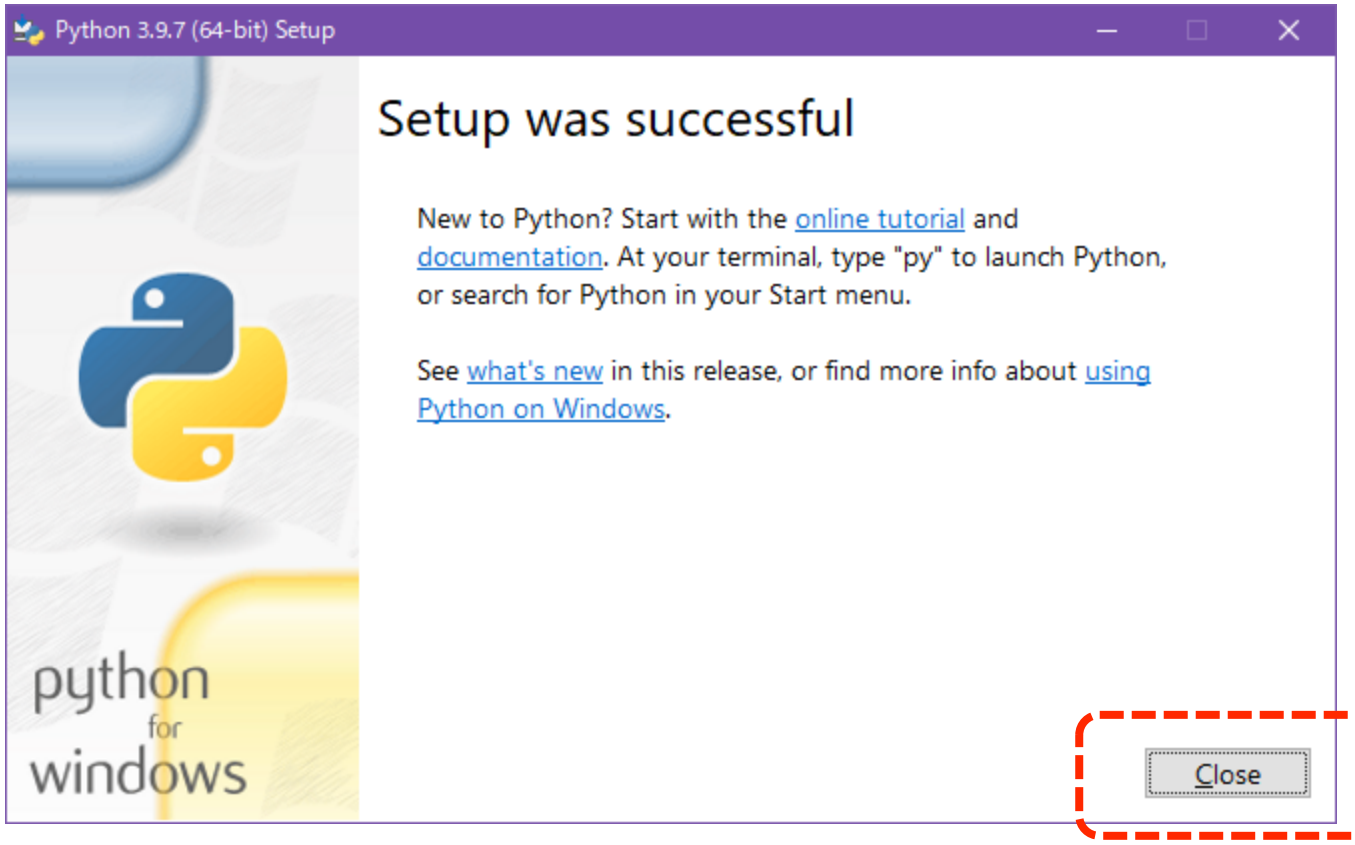
共通

インストールの実施
“python-3.9.7-amd64.exe”の場合
※過去にインストール済みで上位バージョンを
インストールするか初めてインストールする場合



共通

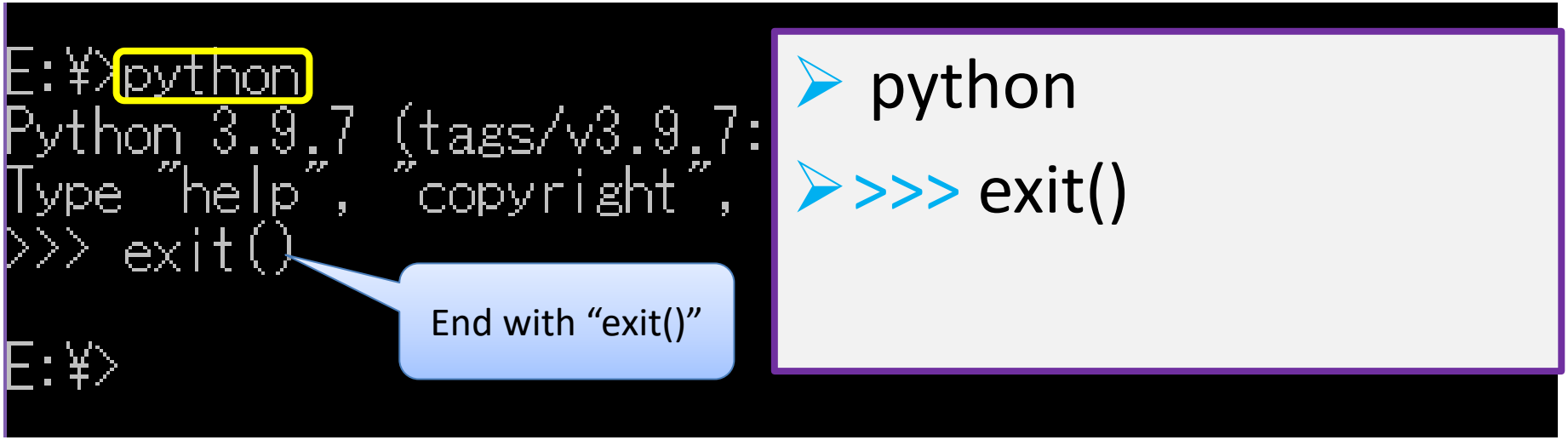
インストール終了


Python 3.9.7 (64-bit) Setup
Setup was successful
New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.
See what's new in this release, or find more info about using Python on Windows.
python for windows
Close

共通

インストール状態の確認

- “cmd.exe” (コマンドプロンプト)を実行
- “python” を入力して実行
- 数行のメッセージ中に表示されるバージョンを確認し、“exit()” と入力して実行し、終了する。



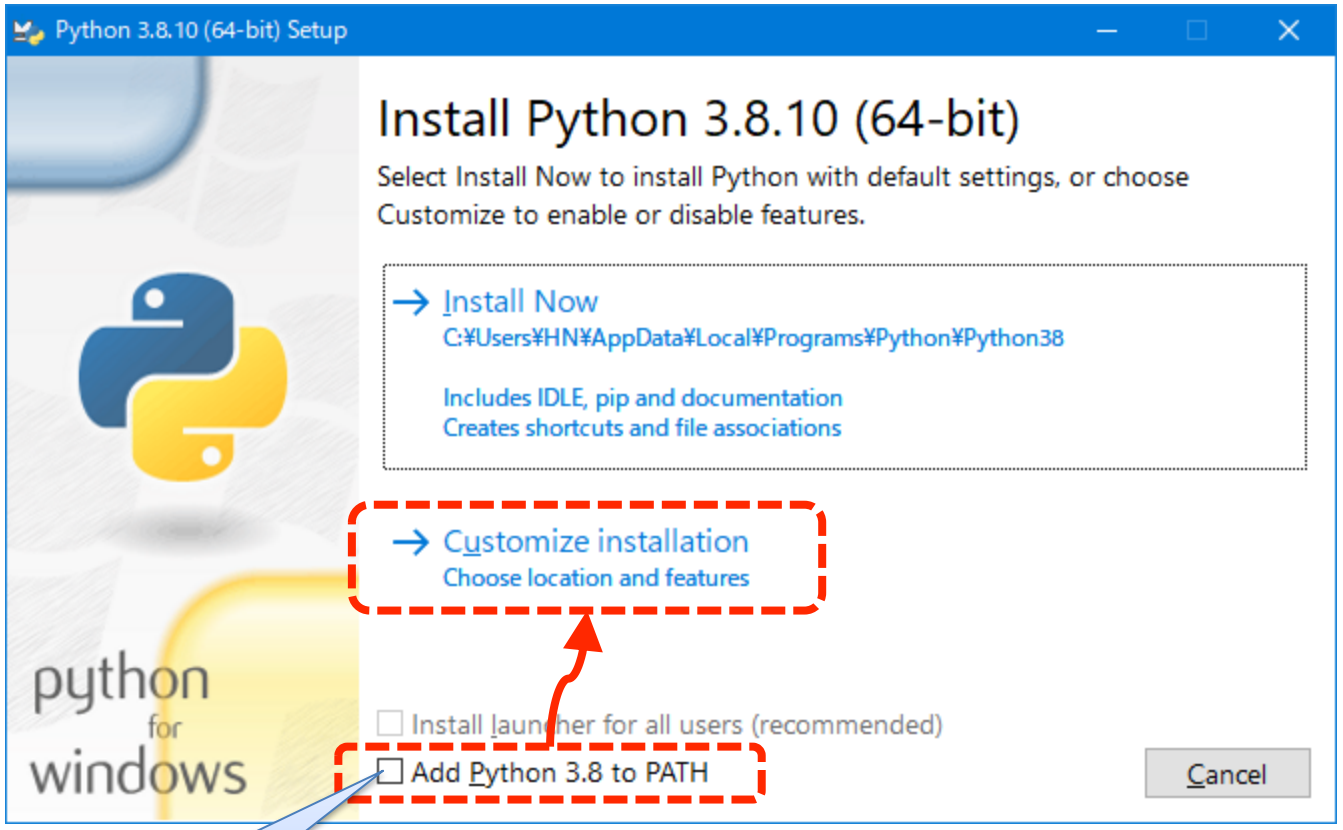
Pythonの特定バージョンのインストールについて

該当バージョン情報の下方
「Files」内から該当ファイルをダウンロード

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		83d71c304acab6c678e86e239b42fa7e	24720640	SIG
XZ compressed source tarball	Source release		d9eee4b20155553830a2025e4dcaa7b3	18433456	SIG
macOS 64-bit Intel installer	macOS	for macOS 10.9 and later	690ddb1be403a7efb202e93f3a994a49	29896827	SIG
macOS 64-bit universal2 installer	macOS	experimental, for macOS 11 Big Sur and later; recommended on Apple Silicon	ae8a1ae082074b260381c058d0336d05	37300939	SIG
Windows embeddable package (32-bit)	Windows		659adf421e90fba0f56a9631f79e70fb	7348969	SIG
Windows embeddable package (64-bit)	Windows		3acb1d7d9bde5a79f840167b166bb633	8211403	SIG
Windows help file	Windows		a06af1ff933a13f6901a75e59247cf95	8597086	SIG
Windows installer (32-bit)	Windows		b355cfc84b681ace8908ae50908e8761	27204536	SIG
Windows installer (64-bit)	Windows	Recommended	62cf1a12a5276b0259e8761d4cf4fe42	28296784	SIG

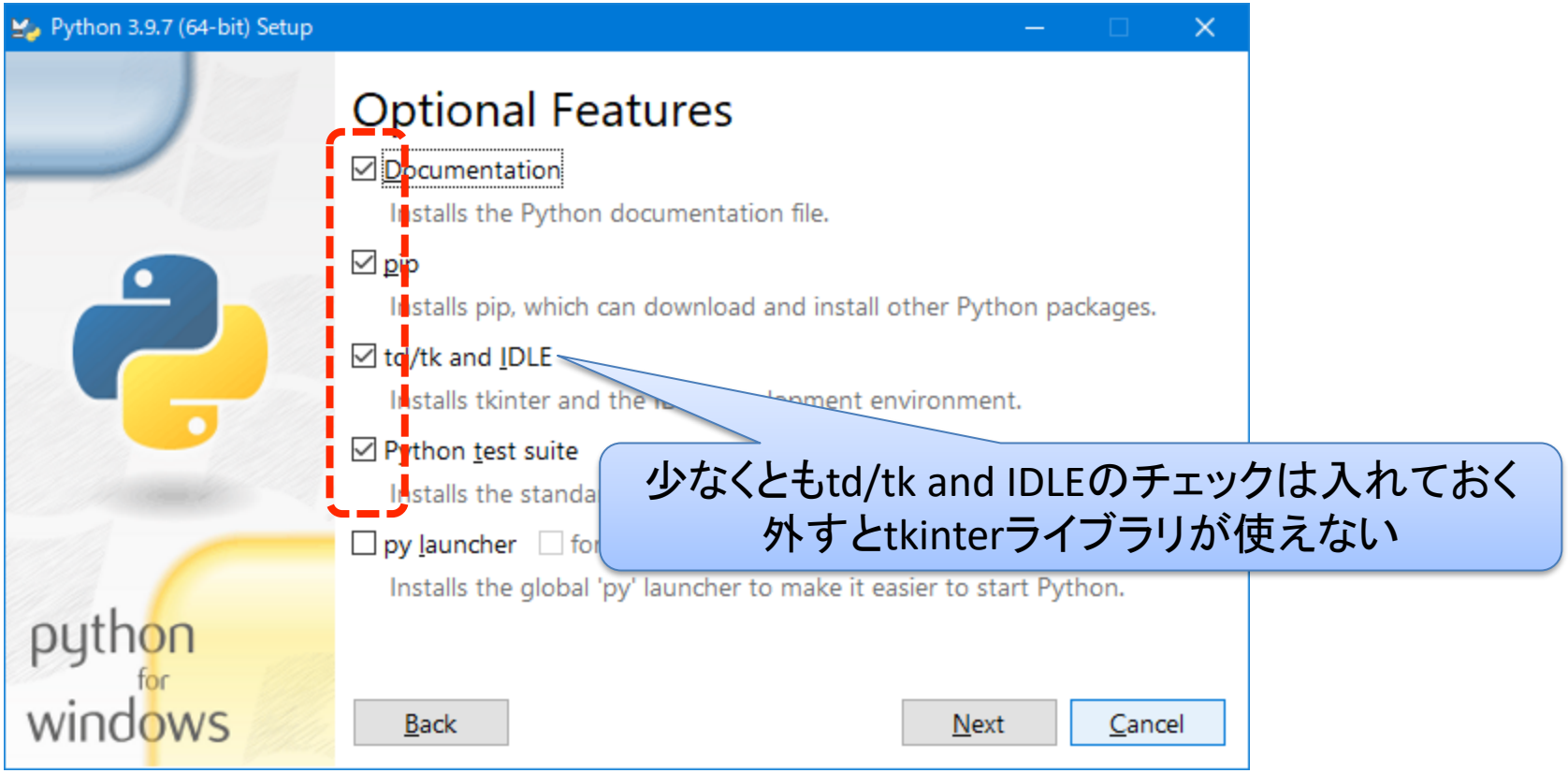
他のPythonのバージョンを追加でインストールする場合（1）



チェックを外す

Pipenvを使って複数のPythonを利用する場合、あらかじめ該当のPythonがインストールされている必要がある。
※pipenv以外のvenv環境を利用する方法では不要な場合もある。

他のPythonのバージョンを追加でインストールする場合（2）

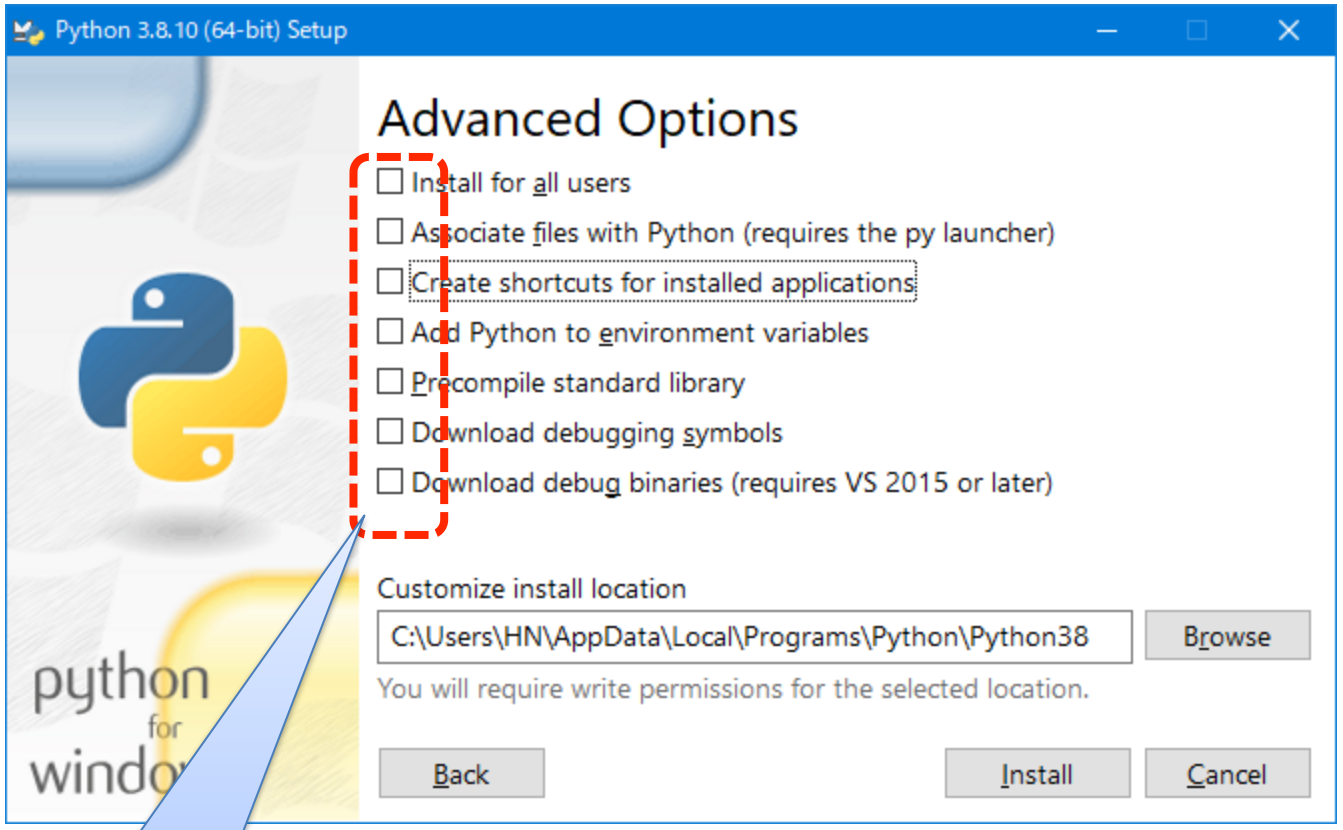


参考資料

https://qiita.com/aujinen/items/b6f7ce3b727cd56bebb8

Juliaにてpipenv仮想環境利用でPyPlot描画できず、結果的にpythonのtkinterインストールの罠に・・・

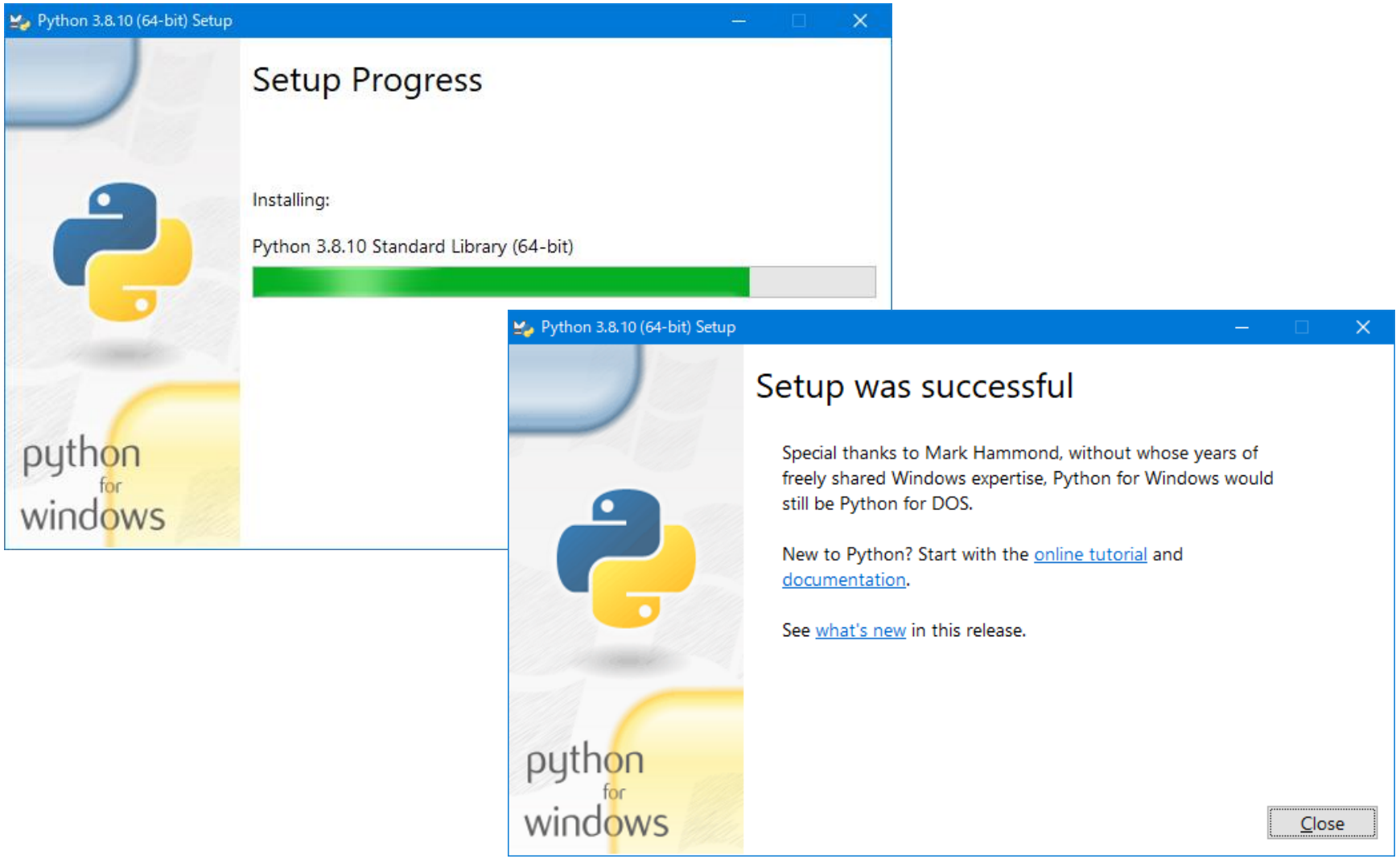
他のPythonのバージョンを追加でインストールする場合（3）



チェックを全て外す

現行のPythonからのアップグレードの場合には、add python to environment variablesにチェックを入れて、該当バージョンのPythonへのリンクを環境変数に埋め込んでおく。

他のPythonのバージョンを追加でインストールする場合（4）



共通

pipのアップグレード

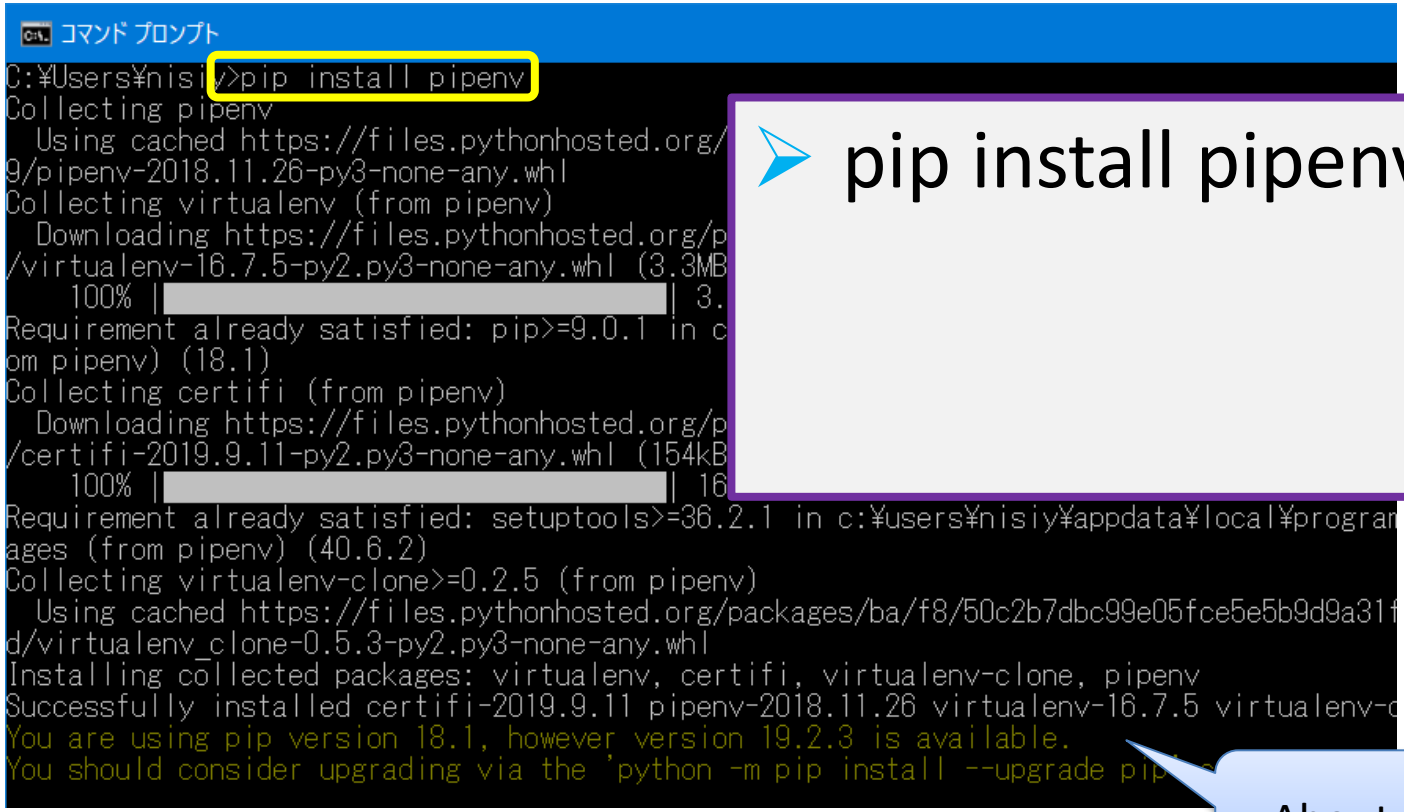
- python -m pip install --upgrade pip

```
E:¥>python -m pip install --upgrade pip
Requirement already satisfied: pip in c:¥users¥hn¥a
Collecting pip
  Downloading pip-21.2.4-py3-none-any.whl (1.6 MB)
     |                                        | 1.6 MB 2.2
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 21.2.3
    Uninstalling pip-21.2.3:
      Successfully uninstalled pip-21.2.3
Successfully installed pip-21.2.4
```

共通

初めてpipenvを使う場合

- pipenvをpipを使ってインストール



共通

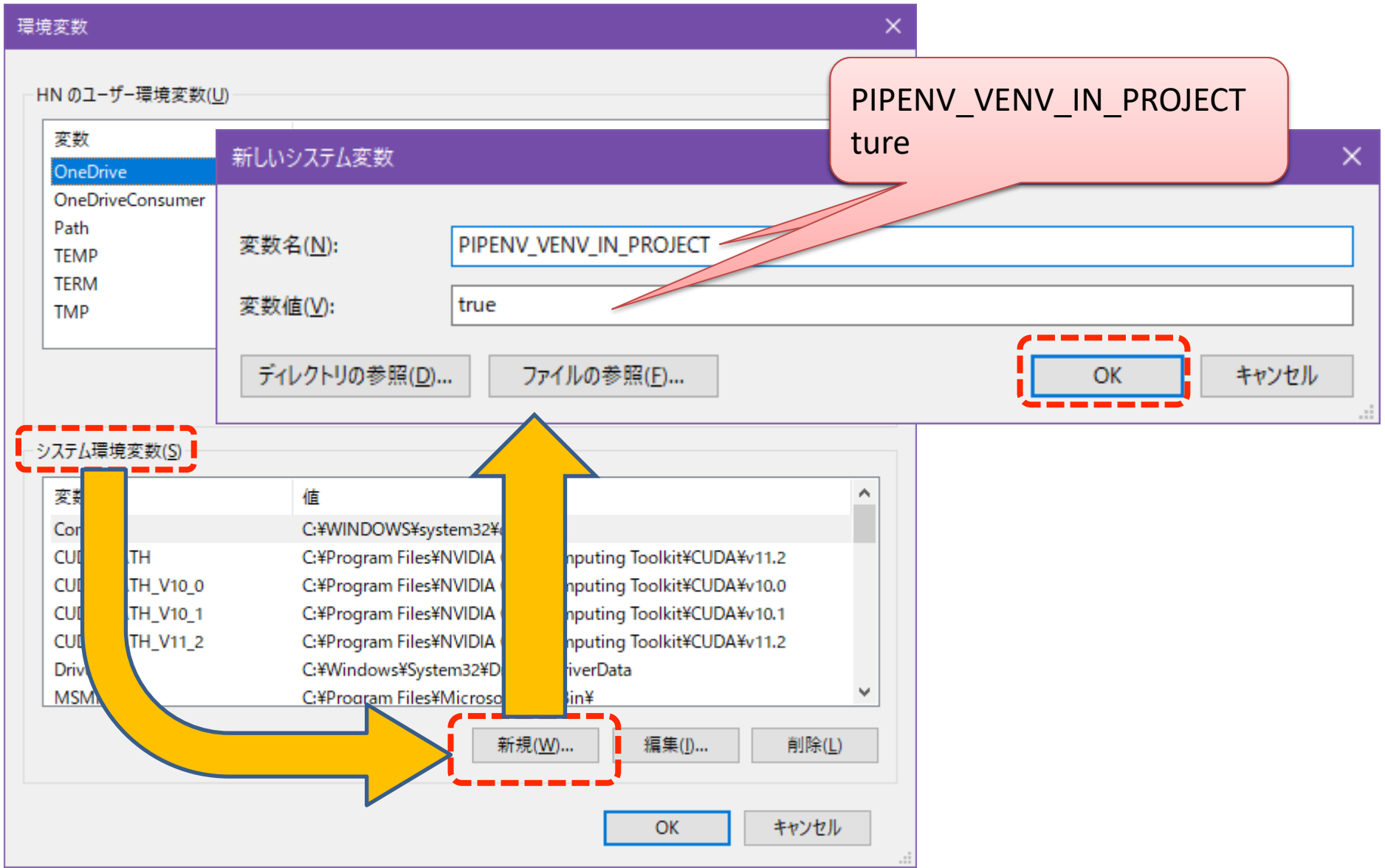
既にpipenvを使っている場合
念のため、pipenvをアップデートする

- pip install -U pipenv

共通

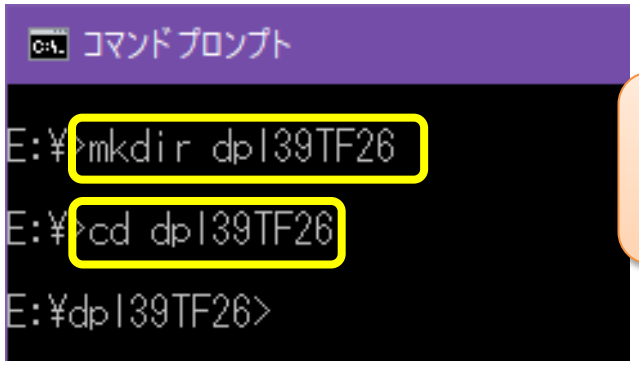
pipenvの環境変数設定

.venvという仮想用フォルダを作業領域内部に作る設定

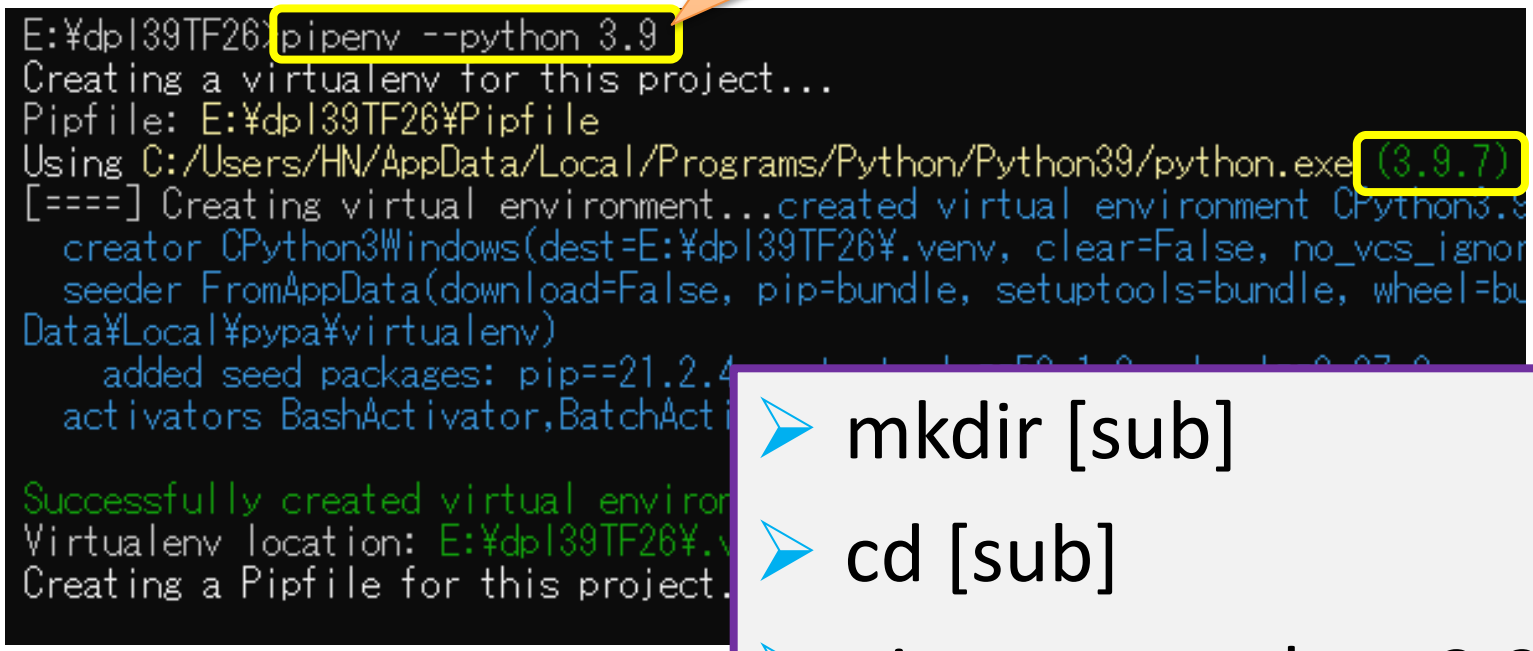


共通

サブディレクトリを作成して移動し、“pipenv install”を実施



- python 3.9.7をインストールするには、
「pipenv --python 3.9」ないし
「pipenv --python 3.9.7」を実行。



- mkdir [sub]
- cd [sub]
- pipenv -- python 3.9

TensorFlow-GPU関連の Pipenv仮想環境構築

※PyTorch関連のPipenv仮想環境構築は
下記PDFを参照して下さい。

https://www5.dent.niigata-u.ac.jp/~nisiyama/grad/Pipenv-PyTorch.pdf

GPU有

GPU利用時 tensorflow-gpuのインストール

- 例えば、2.6.0をインストールする場合

➢ pipenv install tensorflow-gpu==2.6.0

```
E:¥dpl39TF26>pipenv install tensorflow-gpu==2.6.0
Installing tensorflow-gpu==2.6.0...
Adding tensorflow-gpu to Pipfile's [packages]...
Installation Succeeded
Pipfile.lock not found, creating...
Locking [dev-packages] dependencies...
Locking [packages] dependencies...
 Locking...Building requirements...
Resolving dependencies...
Success!
Updated Pipfile.lock (424499)!
Installing dependencies from Pipfile.lock (424499)...
  ================================ 0/0 - 00:00:00
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
```

GPU有

```
E:¥dpl39TF26>pipenv run pip list
Package                      Version
---------------------------- ---------
absl-py                      0.14.1
astunparse                   1.6.3
cachetools                   4.2.4
certifi                      2021.5.30
charset-normalizer           2.0.6
clang                        5.0
flatbuffers                  1.12
gast                         0.4.0
google-auth                  1.35.0
google-auth-oauthlib         0.4.6
google-pasta                 0.2.0
grpcio                       1.41.0
h5py                         3.1.0
idna                         3.2
keras                        2.6.0
Keras-Preprocessing          1.1.2
Markdown                     3.3.4
numpy                        1.19.5
oauthlib                     3.1.1
opt-einsum                   3.3.0
pip                          21.2.4
protobuf                     3.18.0
pyasn1                       0.4.8
pyasn1-modules               0.2.8
requests                     2.26.0
requests-oauthlib            1.3.0
rsa                          4.7.2
setuptools                   58.1.0
six                          1.15.0
tensorboard                  2.6.0
tensorboard-data-server      0.6.1
tensorboard-plugin-wit       1.8.0
tensorflow-estimator         2.6.0
tensorflow-gpu               2.6.0
termcolor                    1.1.0
typing-extensions            3.7.4.3
urllib3                      1.26.7
Werkzeug                     2.0.1
wheel                        0.37.0
wrapt                        1.12.1
```

tensorflow-gpuをインストール後の状況
- pipenv run pip list

でリストアップしたもの

tensorflow-gpuをインストールすると
自動的に
h5py
keras
numpy
がインストールされている。

GPU有

GPUを利用する場合の必要なライブラリ

約1.5GB

- tensorflow-gpu
- numpy *
- keras*
- h5py *

済

- jupyter
- matplotlib
- pandas

- pillow
- scikit-learn
- opencv-python
- scipy
- cython
- pydicom

※「tensorflow」をインストールすると、numpy, keras, h5pyが自動的にインストールされる。

「keras」のバージョンが「2.2.4」を超える場合は、古いPythonソースコードの「import keras」を「from tensorflow import keras」に変更し、また「from keras.XXX import XXX」を「from tensorflow.keras.XXX import」に変更する必要がある。

https://www.tensorflow.org/guide/keras

GPU有

“pipenv install *[ライブラリ名]*” コマンドでのライブラリのインストール

済
- pipenv install tensorflow-gpu ==2.6.0

ライブラリの特定バージョンをインストールする方法

- pipenv install jupyter
- pipenv install matplotlib pandas
- …
- pipenv install pydicom

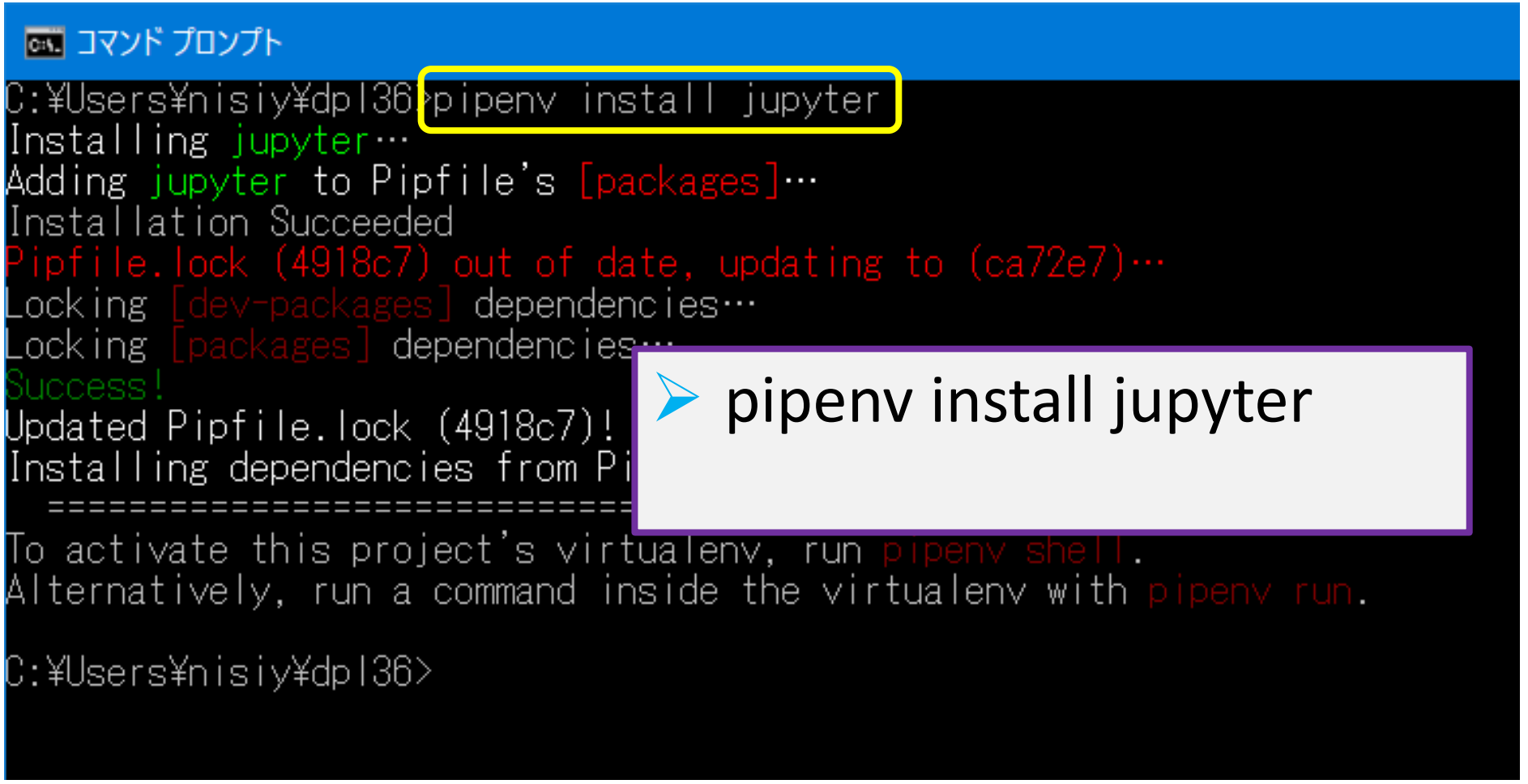
jupyter以降、順にインストールする

複数のライブラリ名を1行に配置することにより、それらを同時にインストールできる。

エラーが発生した場合は、最後の2ページを参照してください。

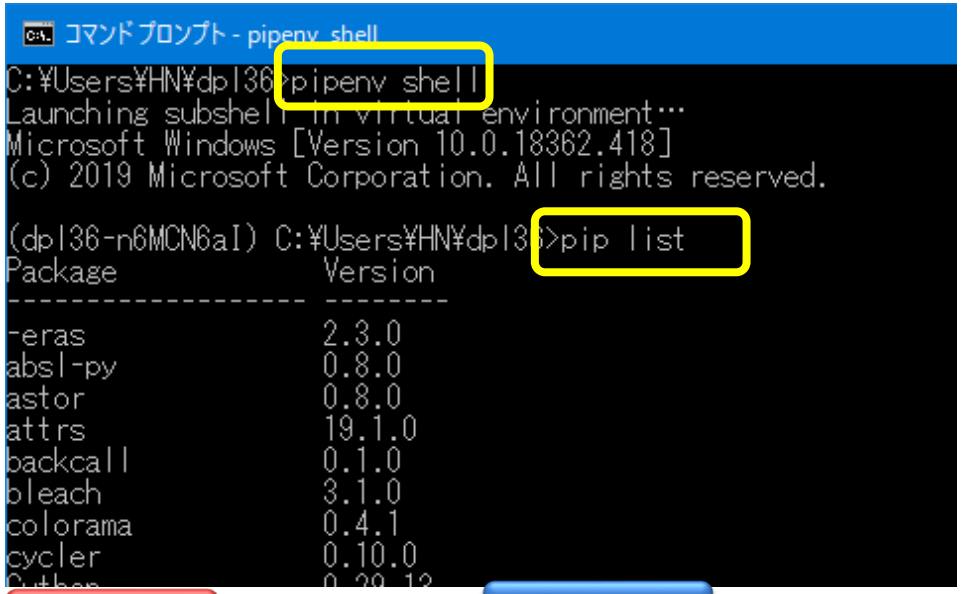
共通

例, “jupyter” ライブラリの “pipenv”環境へのインストール

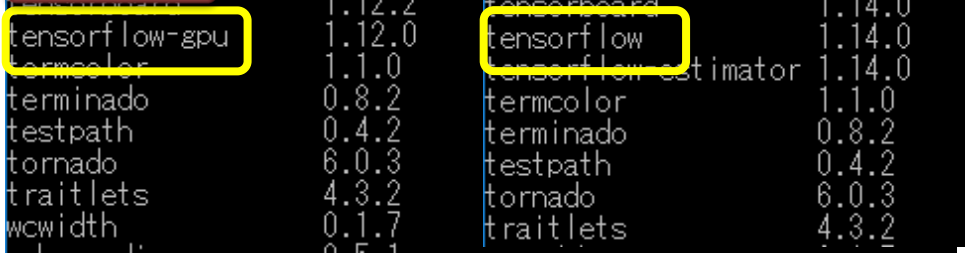


共通

インストールし終えた ライブラリの一覧確認方法



GPU有



GPU無

該当仮想環境を構築したディレクトリに入っている状態で

- pipenv shell

にて、仮想環境を起動し

- pip list

を実行し、表示されるパッケージのリストを確認する。（インストールした以上のリストが出力されるので注意。）
確認後は、

- exit

コマンドで仮想環境から抜ける。
※仮想環境起動中はプロンプト行頭に（仮想環境名）が付随する。

共通

【注意】protobufライブラリ関連

- protobufライブラリのバージョンが新しいとtensorflowライブラリを読み込んだ直後に下記のエラーが出ることがあります。プロトコルバッファーが大幅更新された影響であり、互換性重視で稼働させる場合protobufを3.19.6に固定する必要があります。

```
In [1]: from tensorflow.python.client import device_lib
        device_lib.list_local_devices()
```

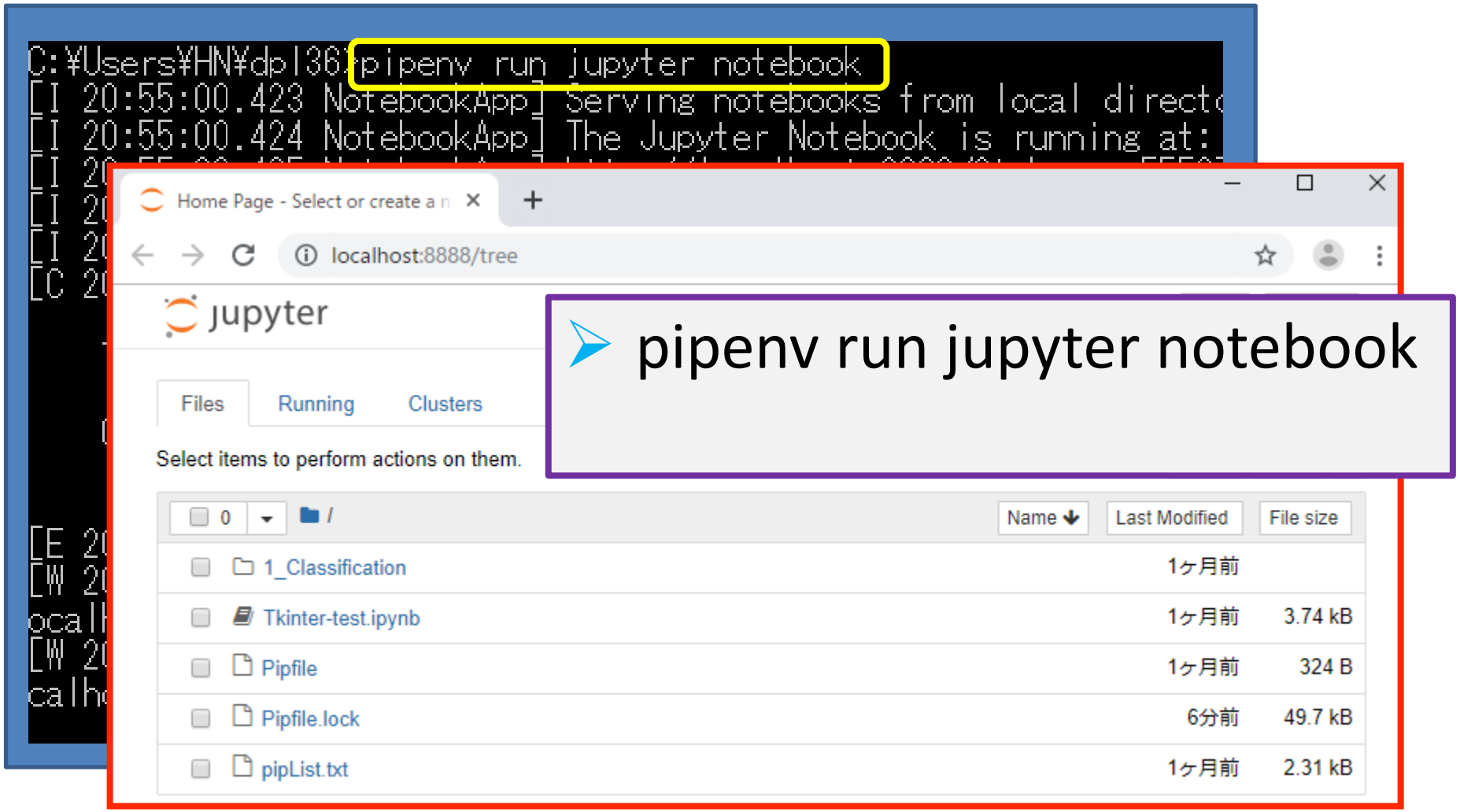
```
    556             default_value, message_type, enum_type, containing_type,
    557             is_extension, extension_scope, options=None,
    558             serialized_options=None,
    559             has_default_value=True, containing_oneof=None, json_name=None,
    560             file=None, create_key=None):  # pylint: disable=redefined-builtin
--> 561   _message.Message._CheckCalledFromGeneratedFile()
    562   if is_extension:
    563     return _message.default_pool.FindExtensionByName(full_name)

TypeError: Descriptors cannot not be created directly.
If this call came from a _pb2.py file, your generated code is out of date and must be regenerated with protoc
>= 3.19.0.
If you cannot immediately regenerate your protos, some other possible workarounds are:
 1. Downgrade the protobuf package to 3.20.x or lower.
 2. Set PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION=python (but this will use pure-Python parsing and will be much
slower).

More information: https://developers.google.com/protocol-buffers/docs/news/2022-05-06#python-updates
```

共通

“pipenv run jupyter notebook” で実行する。



GPU有

TensorFlow-GPU利用時注意点（1）

- Tensorflow-GPU 1.12.0にて、バージョン「2.2.4」の「keras」をインストールできず、「2.2.4」を超える場合は、「keras in tensorflow」（「2.1.6-tf」のバージョン）を使用する必要がある。
- この場合、Pythonソースコードの下記の変更が必要（jupyter notebook内部の置換コマンドで一斉に変更可能）
  - 変換前）import keras
    - 変換後）from tensorflow import keras
  - 変換前）from keras.XXX import YYY
    - 変換後）from tensorflow.keras.XXX import YYY

GPU有

TensorFlow-GPU利用時注意点（2）

- 特殊な変更が必要となるもの
  - 変換前本文中）keras.utils.np_utils.XXX
  - 変換後の本文中）np_utils.XXX
  - 変換後のライブラリ読み込み（冒頭）に追加
    - from tensorflow.python.keras.utils import np_utils

2019/10/22追加

共通

裏技･特定のPipenvの仮想環境を一発でインストールする

- 一度インストール終了したPipenvの仮想環境については、pipfileに設定内容が入っている。
- 設定済みのpipfileを新規の別フォルダにコピーし、同フォルダにて「pipenv install」とすれば、同一の仮想環境が構築される。
- 下記に、「Python=3.10、tensorflow-gpu=2.10.0、protobuf=3.19.6」にて設定したpipfileを置いておきます。
- https://github.com/aujinen/colab_tfbook

GPU有

【注意】TensorFlow-GPU

- WindowsでのTensorFlow-GPUは2.10まで
- 2.11以降のバージョンは未対応。
- Windowsで2.11以降を使いたい場合には、WSL2にて使うこと。

https://www.tensorflow.org/install/source_windows

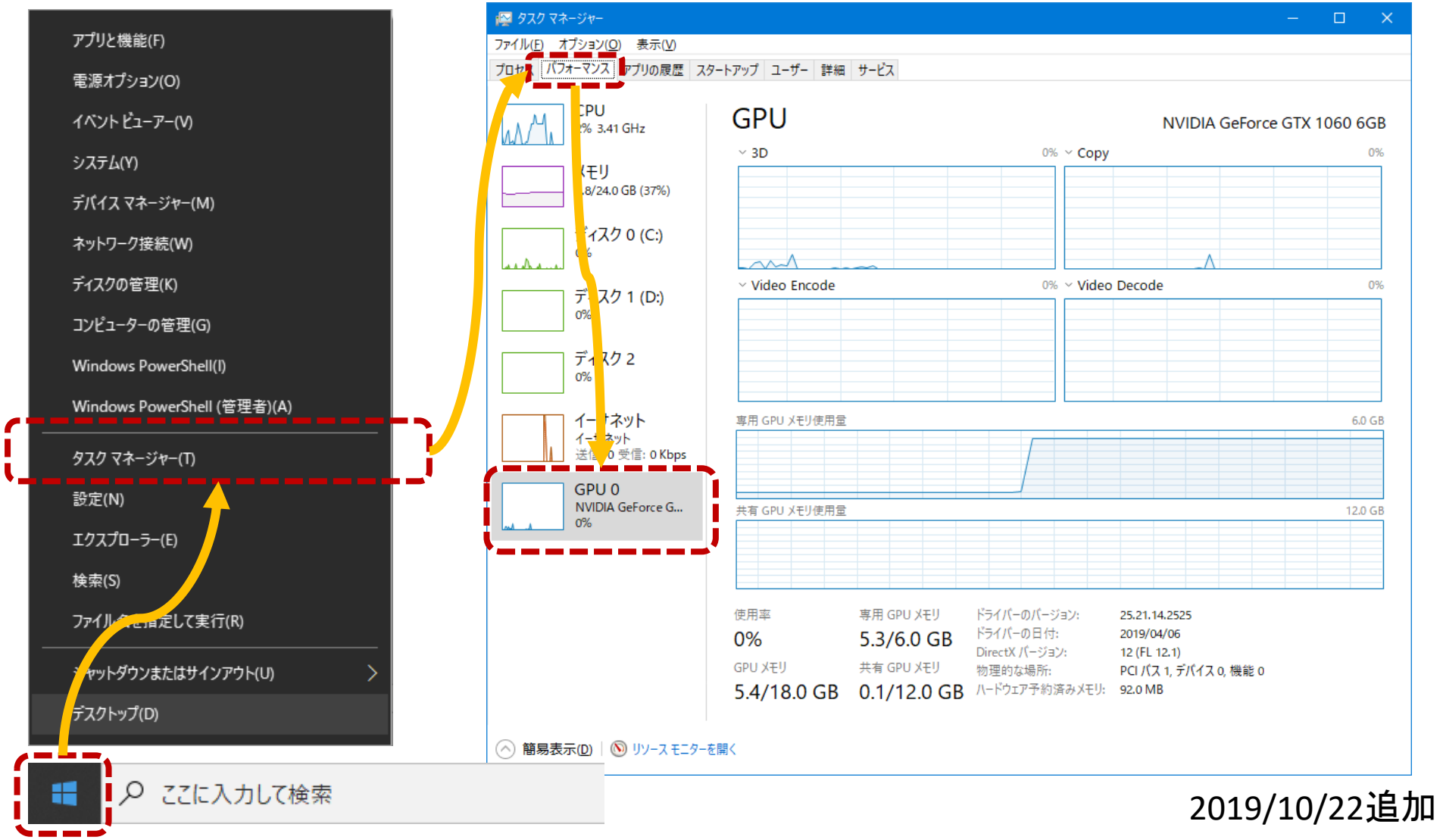
Install GPU support (optional)

See the Windows GPU support guide to install the drivers and additional software required to run TensorFlow on a GPU.

★ **Note:** GPU support on native-Windows is only available for 2.10 or earlier versions, starting in TF 2.11, CUDA build is not supported for Windows. For using TensorFlow GPU on Windows, you will need to build/install TensorFlow in WSL2 or use tensorflow-cpu with TensorFlow-DirectML-Plugin

GPU有

タスクマネージャーでの GPU稼働状況の確認



2019/10/22追加

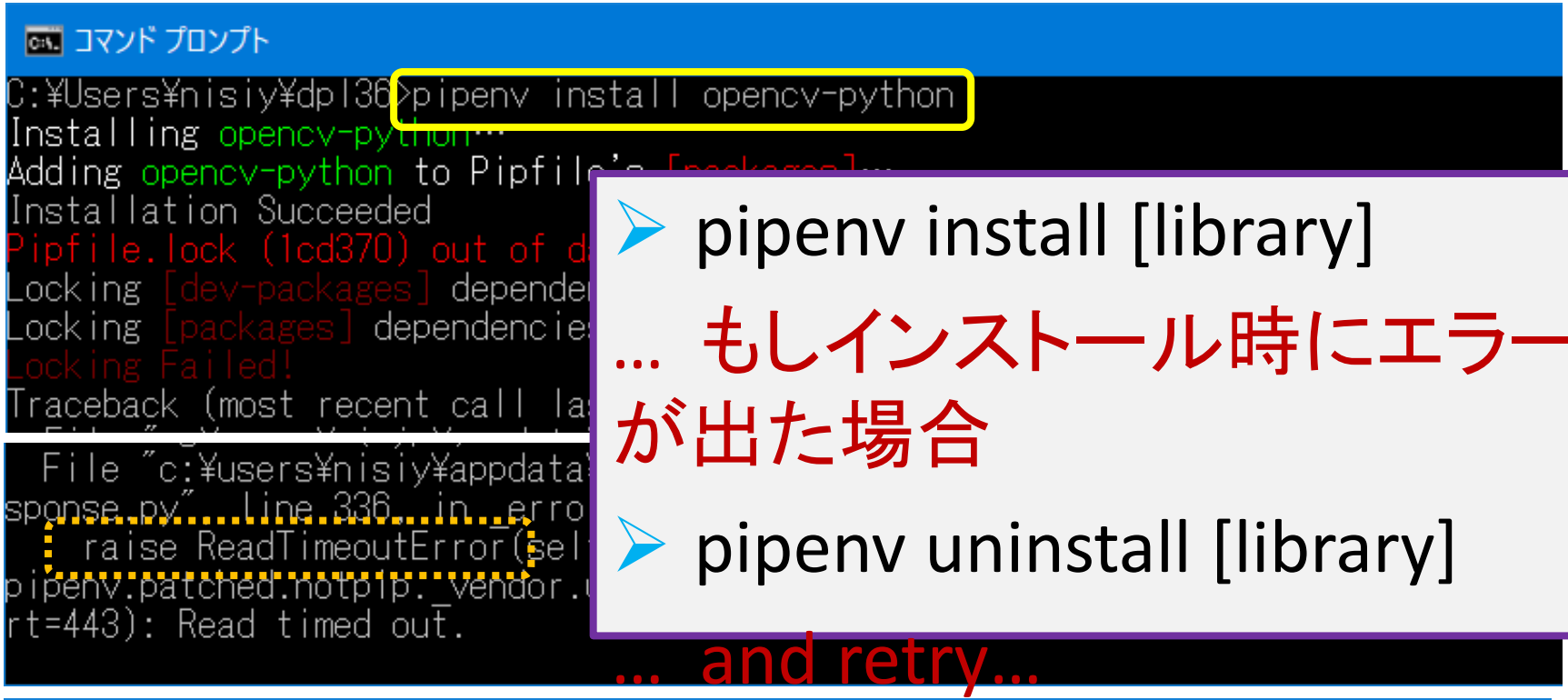
共通

Deep Learningの世界へ続く・・・

(次の2ページはインストールエラー用)

共通

もし、インストール時に下記のようなエラーが出た場合、
ライブラリをアンインストールして、再度インストールしなおす…

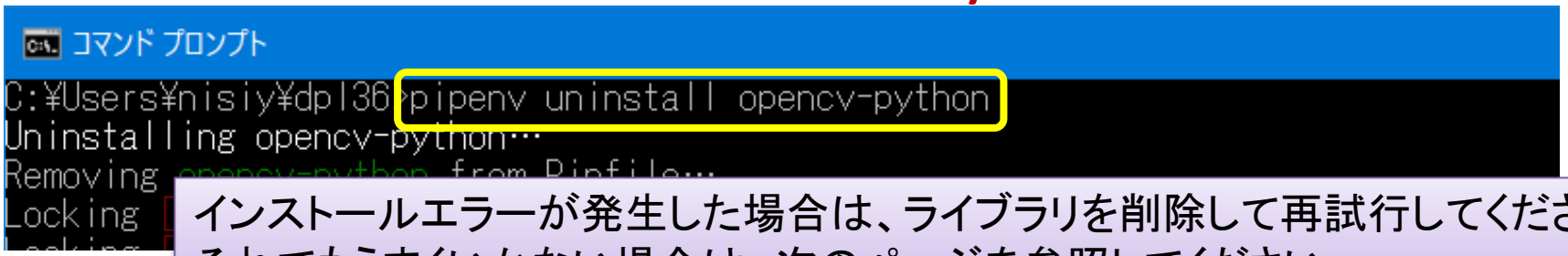


- pipenv install [library]

… もしインストール時にエラーが出た場合

- pipenv uninstall [library]

… and retry…



インストールエラーが発生した場合は、ライブラリを削除して再試行してください。
それでもうまくいかない場合は、次のページを参照してください。

共通

エラー表示の最終行をチェックし、例えば「ReadTimeoutError」であれば、下記のようなサイトを検索して対処

- 英語版
  https://github.com/pypa/pipenv/issues/3110
  - https://github.com/pypa/pipenv/issues/1274
- 日本語版
  - http://pynote.hatenablog.com/entry/python_ubuntu_pipenv#pexpectexceptionsTIMEOUT-%E3%82%A8%E3%83%A9%E3%83%BC
  - https://qiita.com/kazetof/items/b18b225b7a3143514485

Pipenv VS-code関連資料の一部

- pipでアップデートするときのコマンド pip update
  - https://qiita.com/HyunwookPark/items/242a8ceea656416b6da8
- Windowsマシン上でVisual Studio Codeとpipenvを使ってPythonの仮想実行環境を構築する方法（Jupyter notebookも）
  - https://qiita.com/d-kitamura/items/3b8c4ac362668cac6e75
- Pipenvことはじめ
  - https://qiita.com/shinshin86/items/e11c1124e3e2e74556b8

仮想環境でのVS-Code起動方法

- 仮想環境構築したフォルダで
  - code .
  - を実行する。

Jupyter notebookの終了方法
※新しいバージョン