

Pythonによる深層学習

最初に

深層学習(deep learning)を行うにあたって、Pythonを用いることが最も簡単で分かりやすい方法である。

そこでPythonを用いた深層学習の環境構成方法を紹介したいと思う。

それぞれのソフトには相性というものがあり、最新のバージョン同士が必ず動くというわけではない。

したがって2021/3/10時点での当方の環境構成をここに記載する。

Pythonのインストール

Pythonをインストールする。

<https://www.python.org/downloads/release/python-376/>

上記URLからPython 3.7.6

Windows X86-64 executable installerを選択する。

このとき開くインストーラーの中に「PATH(パス)を通す」というチェックボックスが存在した場合、チェックをつける。

Geforceのドライバのダウンロード

深層学習を行うにあたって、グラフィックボード(GPU)と呼ばれる装置を用いている。

GPUは演算処理を行うのが得意で、深層学習を行うにあたって必須の装置である。

また、**GPU**を用いるにあたって様々なソフトをインストールする必要がある。

まず初めにドライバをインストールする。

Geforceのドライバのダウンロード

<https://www.nvidia.co.jp/Download/index.aspx?lang=jp>

上記URLから、PCに搭載しているGPUを選択しダウンロードする。

The screenshot shows the NVIDIA driver download page. At the top, there is a green header with the text "ドライバダウンロード". Below the header, the breadcrumb "NVIDIA > ドライバダウンロード" is visible. On the left side, there is a promotional banner for "GTC 21" with the text "2021年4月12-16日 共に未来を生み出そう AI イノベーター、テクノロジスト、全てのクリエイターのためのカンファレンス" and a "詳細を見る" link. Below the banner, there is another section titled "制作にスピードを". The main content area is titled "NVIDIAドライバダウンロード" and contains the instruction: "下のドロップダウン リストから選択して、お使いの NVIDIA 製品に適したドライバーを特定してください。" followed by a "ヘルプ" link. There are six dropdown menus for selection: "製品のタイプ" (GeForce), "製品シリーズ" (GeForce 10 Series), "製品ファミリー" (GeForce GTX 1080 Ti), "オペレーティングシステム" (Windows 10 64-bit), "ダウンロードタイプ" (Game Ready ドライバー (GRD)), and "言語" (Japanese). A green "検索" button is located below the dropdowns. A question mark icon is present next to the "ダウンロードタイプ" dropdown.

CUDAのダウンロード

もともとゲームグラフィック用に用いられていたGPUを、深層学習などのプログラミングに流用できるように開発されたソフトがCUDAである。

https://developer.nvidia.com/cuda-10.1-download-archive-update2?target_os=Windows&target_arch=x86_64

上記URLからCUDA ver10.1 update2をダウンロードする。
netとlocalは後者を選択する。

Python Release Python 3.7.6 | Py... x CUDA Toolkit 10.1 update2 Arch... x CUDA Compatibility : GPU Depl... x +

developer.nvidia.com/cuda-10.1-download-archive-update2?target_os=Windows&target_arch=x86_64&target_version=10&target_type=exe-local

NVIDIA DEVELOPER HOME BLOG NEWS FORUMS DOCS DOWNLOADS TRAINING ACCOUNT

CUDA Toolkit 10.1 update2 Archive

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System: [Windows](#) [Linux](#) [Mac OSX](#)

Architecture: [x86_64](#)

Version: [10](#) [8.1](#) [7](#) [Server 2019](#) [Server 2016](#) [Server 2012 R2](#)

Installer Type: [exe \(network\)](#) [exe \(local\)](#)

Download Installer for Windows 10 x86_64

The base installer is available for download below.

> **Base Installer** [Download \(2.5 GB\)](#)

Installation Instructions:

1. Double click cuda_10.1.243_426.00_win10.exe
2. Follow on-screen prompts

The checksums for the installer and patches can be found in [Installer Checksums](#).
For further information, see the [Installation Guide for Microsoft Windows](#) and the [CUDA Quick Start Guide](#).

[Documentation >](#) [Release Notes >](#) [Code Samples >](#) [Legacy Releases >](#)

Additional Resources



Experience

検索するには、ここに入力します

CUDA Toolkit 10.1 ...

Microsoft

18:53 2021/03/04

cudaをダウンロード

cudaはCUDAの拡張機能である。

<https://developer.nvidia.com/cuda>

上記URLから「Download cuDNN」を選択後、アカウント作成が求められた場合、作成するかGoogleのアカウントを使うことができるはずである。

Archived cuDNN releasesをクリックして

cuDNN v7.6.5 for CUDA 10.1を選択してダウンロードする。

[Download cuDNN v8.0.5 \(November 9th, 2020\), for CUDA 11.0](#)

[Download cuDNN v8.0.5 \(November 9th, 2020\), for CUDA 10.2](#)

[Download cuDNN v8.0.5 \(November 9th, 2020\), for CUDA 10.1](#)

[Download cuDNN v8.0.4 \(September 28th, 2020\), for CUDA 11.1](#)

[Download cuDNN v8.0.4 \(September 28th, 2020\), for CUDA 11.0](#)

[Download cuDNN v8.0.4 \(September 28th, 2020\), for CUDA 10.2](#)

[Download cuDNN v8.0.4 \(September 28th, 2020\), for CUDA 10.1](#)

[Download cuDNN v8.0.3 \(August 26th, 2020\), for CUDA 11.0](#)

[Download cuDNN v8.0.3 \(August 26th, 2020\), for CUDA 10.2](#)

[Download cuDNN v8.0.3 \(August 26th, 2020\), for CUDA 10.1](#)

[Download cuDNN v8.0.2 \(July 24th, 2020\), for CUDA 11.0](#)

[Download cuDNN v8.0.2 \(July 24th, 2020\), for CUDA 10.2](#)

[Download cuDNN v8.0.2 \(July 24th, 2020\), for CUDA 10.1](#)

[Download cuDNN v8.0.1 RC2 \(June 26th, 2020\), for CUDA 11.0](#)

[Download cuDNN v8.0.1 RC2 \(June 26th, 2020\), for CUDA 10.2](#)

[Download cuDNN v7.6.5 \(November 18th, 2019\), for CUDA 10.2](#)

[Download cuDNN v7.6.5 \(November 5th, 2019\), for CUDA 10.1](#)

[Download cuDNN v7.6.5 \(November 5th, 2019\), for CUDA 10.0](#)

[Download cuDNN v7.6.5 \(November 5th, 2019\), for CUDA 9.2](#)

[Download cuDNN v7.6.5 \(November 5th, 2019\), for CUDA 9.0](#)

[Download cuDNN v7.6.4 \(September 27, 2019\), for CUDA 10.1](#)

[Download cuDNN v7.6.4 \(September 27, 2019\), for CUDA 10.0](#)

[Download cuDNN v7.6.4 \(September 27, 2019\), for CUDA 9.2](#)

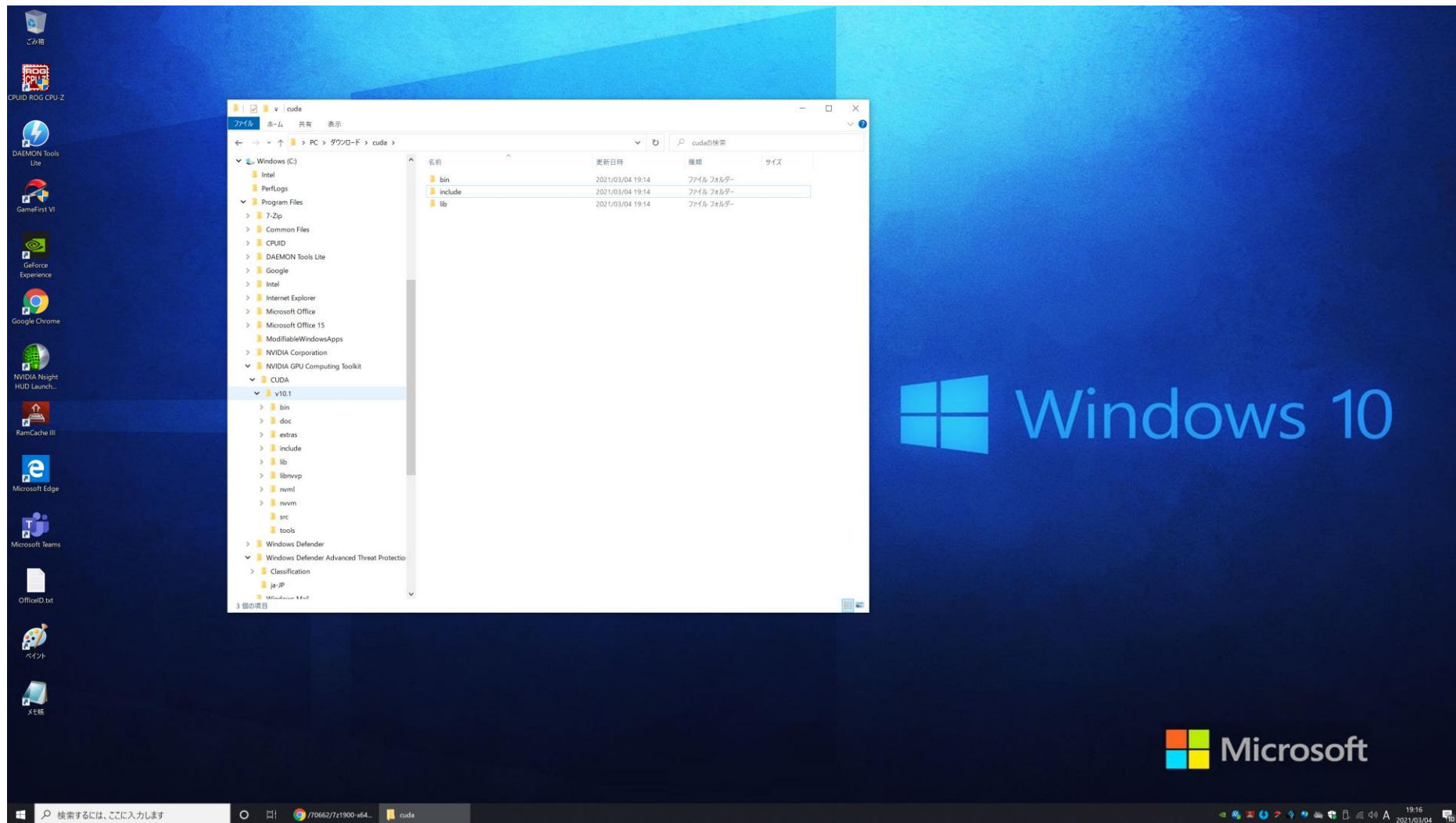
[Download cuDNN v7.6.4 \(September 27, 2019\), for CUDA 9.0](#)

cuDNNのフォルダを移動

cuDNNをダウンロードし、zipファイルを解凍後中に入っているフォルダをCUDAの中にコピーする。

解凍後のファイル「cuda」の中に入っているファイルをすべて
Program Files→Nvidia GPU computing Toolkit→CUDA→v10.1
にコピーする。

ファイルを置き換えるか選択する必要がある場合、すべてのファイルに権限を利用して置き換えるを選択する。



いま開いているフォルダはcuDNNを解凍したものである。
これを左の青く表示されたv10.1というところにコピーする。

画像はコピー後のため3つのファイルが存在しているが、コピー前はテキストファイルも存在はずであるのでそれを含め全てコピーする。

PATHを通す

CUDAに自由にアクセスするためにPATHというものを設定する必要がある。

Windowsマークを右クリック→システム→システム情報
→システムの詳細設定→環境変数

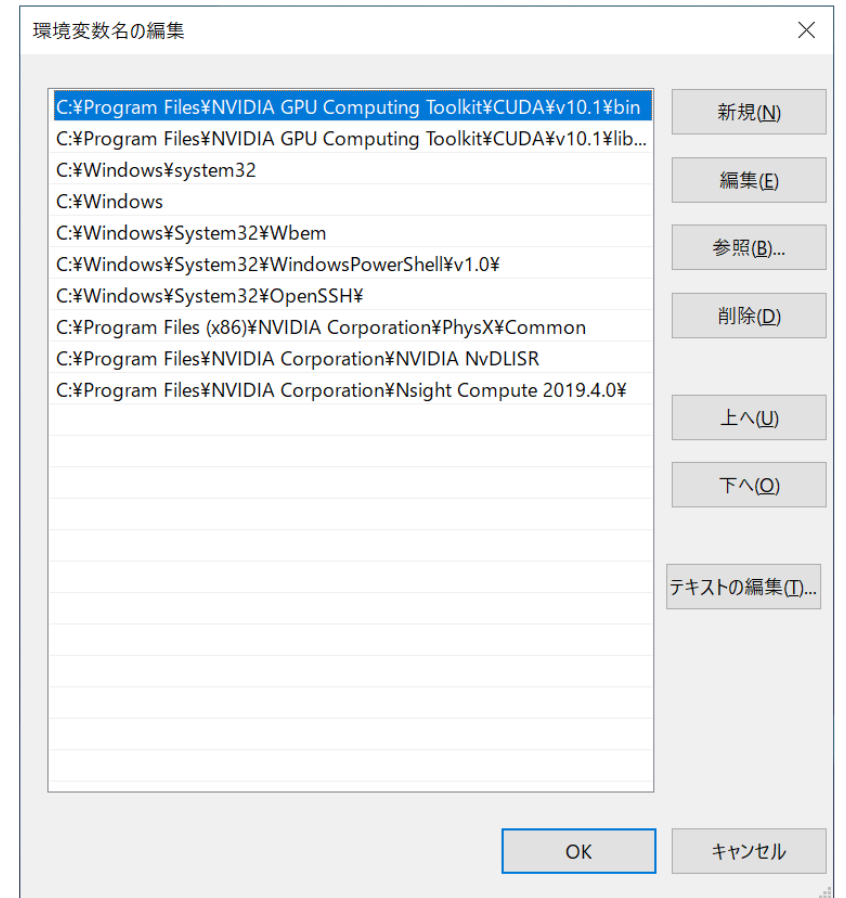
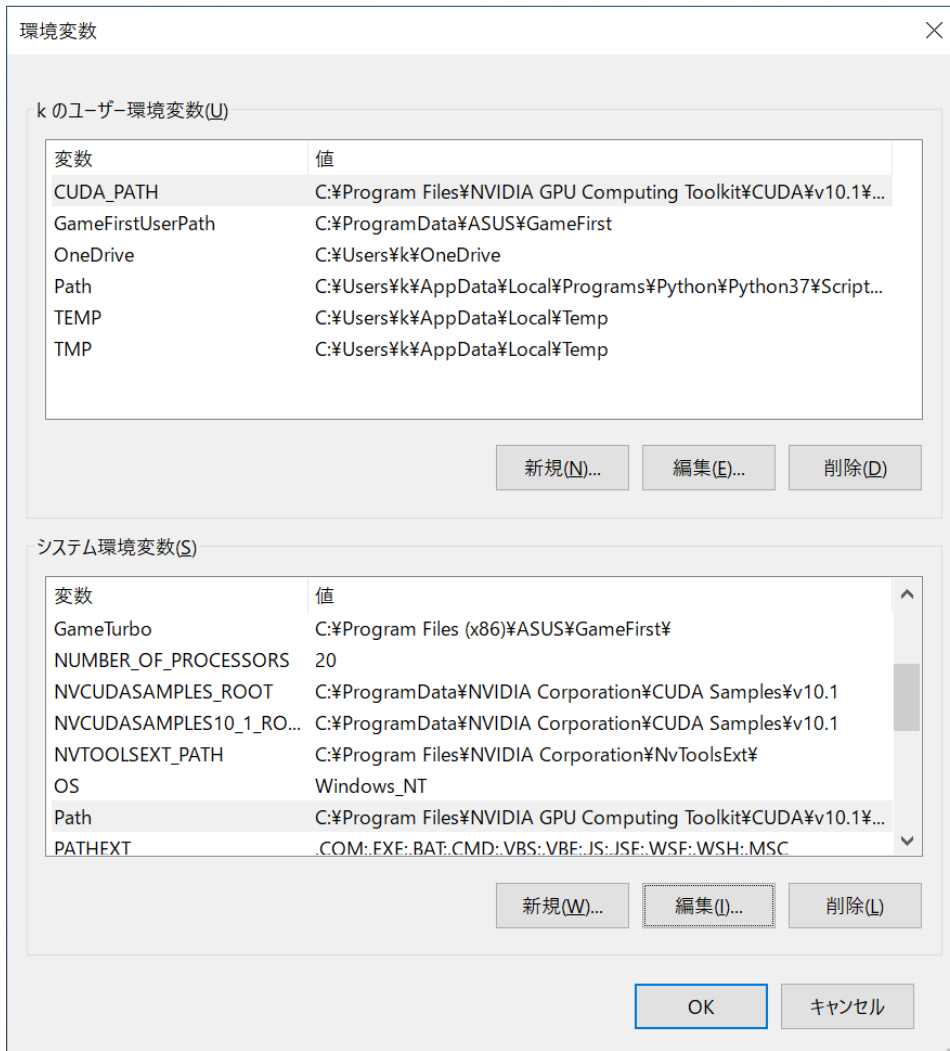
システム環境変数というもののの中に「Path」というものがあるため、それをダブルクリックする。

中に次のスライドに示すように

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1\bin

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1\libnvvp

がはいっているか確認し、なければ追加する。



仮想環境を作成

仮想環境を作成する。仮想環境について特に深く触れないが、深層学習を行うにあたって必須とあって差し支えない。気になる方は加藤研究室まで来訪をお待ちしています。

コマンドプロンプトを開く。

「`pip install virtualenv`」

と入力する。その後

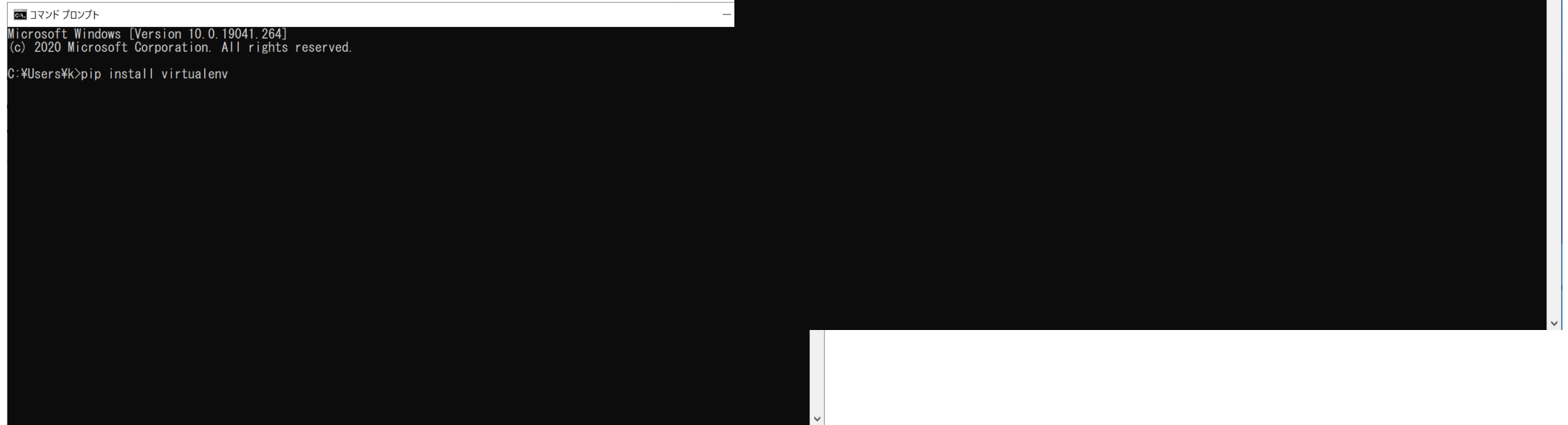
「`python -m venv “任意の名前”`」

と入力する。

ここで赤字でpipのアップデートが必要であると出てきた場合、表示されたコードを打つことでアップデート可能である。

仮想環境を作成

任意の名前は自分の名前+pyがわかりやすいためおすすめである。



```
コマンド プロンプト
Microsoft Windows [Version 10.0.19041.264]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\%k>pip install virtualenv

コマンド プロンプト
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.
C:\Users\%k>python -m venv katoypy_
```

仮想環境の確認

仮想環境を作成したら、作成が完了したことを確認するために仮想環境を起動する。

コマンドプロンプトで

「“仮想環境の名前”¥Scripts¥activate」

と入力し、起動することを確認する。

終了する際は

「exit()」と入力する。


```
コマンド プロンプト
Microsoft Windows [Version 10.0.19041.264]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\%k>takagipy%Scripts%activate
```

```
コマンド プロンプト
(takagipy) C:\Users\%k>exit()
```

VScodeをダウンロード

プログラミングコードを書いてそれを実行するアプリケーションをインストールする必要がある。

そこで当方のおすすめであるVScodeのダウンロードについて記述する。VScodeは拡張機能が豊富であるため、自分にあったカスタマイズが可能である。

<https://azure.microsoft.com/ja-jp/products/visual-studio-code/>

上記URLからダウンロードを行う。

VScodeの設定

VScodeを開いたら拡張機能から「japanese」と検索してインストールすることで日本語化することができる。

次に「python」と検索し、インストールする。

ここまで終了すると、一度VScodeを閉じてもう一度開く。

VScodeの設定

歯車の設定から、「path」と検索する。

Python: Default Interpreter Pathというところに
C:¥Users¥ユーザー名¥仮想環境の名前¥Scripts
と入力する。

Python: Env Fileというところに
C:¥Users¥ユーザー名¥仮想環境の名前
と入力する。

Python: Venv Pathというところに
C:¥Users¥ユーザー名
と入力する。

Vscodeの設定

The screenshot shows the Visual Studio Code settings interface for Python. The left sidebar displays the file explorer with the 'test.py' file selected. The main settings pane is titled 'path' and lists various Python-related settings:

- Python > Auto Complete: Typed Paths**: Specifies paths to local typeshed repository clone(s) for the Python language server. Includes a button '項目の追加'.
- Python: Conda Path**: Path to the conda executable to use for activation (version 4.4+).
- Python: Default Interpreter Path**: Path to Python, you can use a custom version of Python by modifying this setting to include the full path. Value: C:\Users\k\takagipy\Scripts.
- Python: Env File**: Absolute path to a file containing environment variable definitions. Value: C:\Users\k\takagipy.
- Python > Formatting: Autopep8 Path**: Path to autopep8, you can use a custom version of autopep8 by modifying this setting to include the full path. Value: autopep8.
- Python > Formatting: Black Path**: Path to Black, you can use a custom version of Black by modifying this setting to include the full path. Value: black.
- Python > Formatting: Yapf Path**: Path to yapf, you can use a custom version of yapf by modifying this setting to include the full path. Value: yapf.
- Python: Jedi Path**: Path to directory containing the Jedi library (this path will contain the 'Jedi' sub directory). Note: since Jedi depends on Parso, if using this setting you will need to ensure a suitable version of Parso is available. This setting only works with 'languageServer=Jedi' and not Jedi.LSP.

The bottom of the window shows a terminal window with the following content:

```
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォームの PowerShell をお試しください https://aka.ms/pscore6

PS C:\Users\k\takagipy>
```

The status bar at the bottom indicates the current Python environment: Python 3.7.6 64-bit (takagipy).

VScodeの設定

終了後、一度閉じて再起動を行う。

その後、**open file**(またはファイルを開く)から自分の仮想環境を選択する。

もう一度再起動すると画面左側に仮想環境名が表示される。

仮想環境名の右にある新しいファイルアイコンをクリックして“**test.py**”などの新しいファイルを作成すると、左下の部分に**Python3.7.6**と仮想環境の名前が表示されれば成功である。

※コードを書くファイル名には拡張子として“**.py**”が必要

ライブラリ

プログラミングにはライブラリと呼ばれる便利アイテムがあるが、自分自身でインストールする必要があるため、必要に応じてインストールする。

例えば...

- matplotlib → 図のプロット
- pandas → 表作成

インストールする場合はコマンドプロンプトで

「“仮想環境の名前”¥Scripts¥activate」と入力して仮想環境を起動する。

その後、

「pip install ライブラリ名」

とするとインストールが開始する。

最後に

何か質問がありましたら、加藤研究室への来訪をお待ちしております。

みなさんの来訪をお待ちしております。

医理工学院修士2年

高木耀介