

学習環境の構築方法

1 プログラミングを学習するための機器とアプリのインストール、設定

プログラミングを学習するために用意した機器は、A タイプ、B タイプ、C タイプの 3 種類です。1 種だけを選んで運営することもできますし、いくつか選んで運営することもできます。

ただし、テキスト「はじめてのプログラミング～Python 編」は、**学習環境が設定済み**を前提としています。こちらの「1 プログラミングを学習するための機器とアプリのインストール、設定」は、運営側が事前に実施していただきますようお願いいたします。ご注意ください。

3 種の機器には次のような違いがあります。

- A タイプ : **Windows10 一般型**

→一般的なインストールにより学習環境を構築するタイプ

Windows10 パソコンがあり、OS の環境変数変更の問題がない方、職業能力開発施設に向いています。

- B タイプ : **Windows10 WinPython 活用型**

→この中で最も学習環境構築が簡単なタイプ

Windows10 パソコンがあり、OS の環境変数の変更をせず、Python や Visual Studio Code を導入したい方、職業能力開発施設に向いています。WinPython は A タイプの Python に比べ処理速度が若干遅いものの Python の実行環境だけでなく、主要なライブラリ群や Visual Studio Code など開発環境も一括で導入してくれます。ポータブル化されていますので、OS の環境変数に書き込むことはありません。設定した WinPython のフォルダを丸ごとほかのコンピュータにコピー・貼り付けすれば使用できるようになります。USB メモリに保存すれば異なる Windows10 パソコンでも同じ学習環境でプログラミングができます。DVD-R などに設定した WinPython のフォルダをコピーし、授業用のパソコンに貼り付ければ、授業準備を省力化することができます。

- C タイプ : **組み込みコンピュータ Raspberry Pi 活用型**

→この中で最も機器にかかる費用が低いタイプ

テレビとキーボード、マウスはあるものの、プログラミングのために活用できるパソコンがなく、低い費用でのプログラム開発環境導入を検討している方、職業能力開発施設に向いています。Raspberry Pi の USB ポートにキーボードとマウスをつなぎ、テレビを HDMI ケーブルでつなげば、デスクトップ・パソコンと同等のコンピュータになります。さらに、接続可能なインターネット回線があれば、有線あるいは無線で接続できますので、さらに様々な活用ができることでしょう。

A タイプ Windows10 パソコン

OS : Windows10, Apps : Python3.7+Visual Studio Code

【ハードウェア】

Windows10 のパソコン

【ソフトウェア】

OS : Windows 10 Pro 1909 (Windows 10 Home 1909)

アプリ : Python3.7.7 64bit

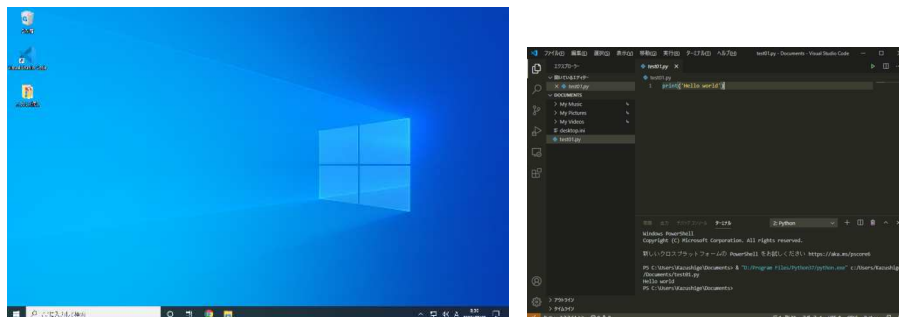
アプリ : Visual Studio Code1.46.1

アプリ : Google Chrome

設定内容

アプリ : Python3.7.7 64bit をインストールします。
冒頭の ☒ Add Python 3.7 to Path にチェックを付けておきます。

アプリ : Visual Studio Code1.46.1 をインストールします。
(スタートメニューから Visual Studio Code を見つけ、 **ファイルの場所を開く** をクリックし、ショートカットをコピー、デスクトップに貼り付けます。)



Visual Studio Code のメニューを **日本語化** するために、Visual Studio Code を起動し、メニュー **view > Extensions** をクリックします。
入力欄に **python** と打ち込み、現れた Python の **[install]** をクリックします。
入力欄に **Japanese** と打ち込むと
(Japanese Language Pack for Visual Studio Code が現れるので)
Japanese Language Pack for Visual Studio Code の **[install]** をクリック
します。右下に出る青い **[Restart now]** をクリックすると再起動され、日本語化を確認
できます。

Visual Studio Code から **Python** を実行するために、メニュー **表示 > コマンド
パレット** をクリックし、**select** と入力します。表示された中にある **[Python: インタ
プリター]** を選択し、**Python3.7.7 64bit** を選択します。これで Visual
Studio Code で編集している Python プログラムが Visual Studio code から
実行できるようになります。
また、メニュー **ファイル > ユーザー設定 > 配色テーマ > monokai** にします。
☐ 起動時にウェルカムページを表示する のチェックを外します。
右下に出てくる通知の **[x]** をクリックし、表示を消します。([Don't show again])

アプリ : Google Chrome もインストールしておきます。

ファイル : **ファイルの入った「入力ミス探し」フォルダをデスクトップに保存しておきます。**

B タイプ Windows10 パソコン
OS : Windows10, Apps : WinPython64-3.7cod

【ハードウェア】

Windows10 のパソコン

【ソフトウェア】

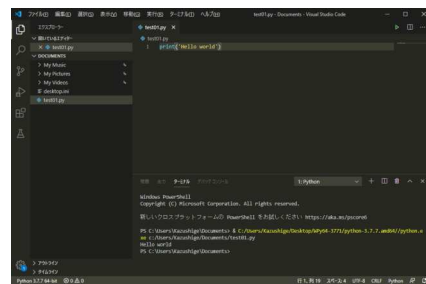
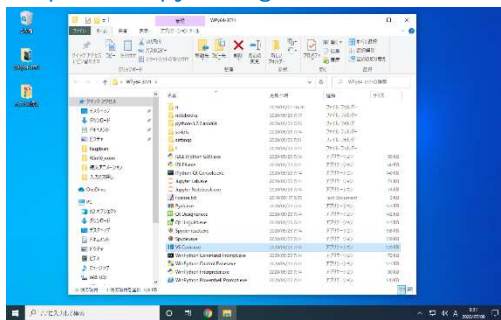
OS : Windows 10 Pro 1909 (Windows 10 Home 1909)
アプリ : WinPython_Code64-3.7.7.1cod (Python3.7 と Visual Studio Code)
アプリ : Google Chrome

設定内容

アプリ : WinPython_Code64-3.7.7.1cod ファイルの書き込み
(Python3.7 と Visual Studio Code)

次のリンクより Winpython64-3.7.7.1cod.exe ファイルをダウンロードし、実行すると展開場所を尋ねてくるので (Extract to:), **デスクトップ** を指定し **[Extract]** ボタンをクリックします。「**WPy64-3771**」という名前のフォルダがデスクトップに作られ、Python や Visual Studio Code などのファイルがそのフォルダの中に書き込まれます。

<https://winpython.github.io/>



Visual Studio Code の **日本語化** をするために、**VS Code.exe** を起動し、メニュー **view > Extensions** をクリックします。
入力欄に **python** と打ち込み、現れた Python の **[install]** をクリックします。
入力欄に **Japanese** と打ち込むと
(Japanese Language Pack for Visual Studio Code が現れるので)
Japanese Language Pack for Visual Studio Code の **[install]** をクリックします。右下に出る青い **[Restart now]** ボタンをクリックすると再起動され、日本語化を確認できます。

また、メニュー **ファイル > ユーザー設定 > 配色テーマ > monokai** にします。
☐ 起動時にウェルカムページを表示する のチェックを外します。
右下に出てくる通知の **[x]** をクリックし、表示を消します。([Don't show again])

アプリ : Google Chrome もインストールしておきます。

ファイル : **ファイルの入った「入力ミス探し」フォルダをデスクトップに保存しておきます。**

Cタイプ Raspberry Pi4 Model B 2GB

OS : Raspbian buster, Apps : Python3+Visual Studio Code

【ハードウェア】

Raspberry Pi4 2GB 版

[参考] ラズパイ4スターターキット 2GB版[AE-RASPI4-STK(2GB)]

(秋月電子通商 : ￥7,700 2020/07/02 現在)

①～⑤が同梱されています。

①M-14839 Raspberry Pi4 Model B 2GB

②P-14781 RASPBERRY Pi4 STANDARD CASE (3 ピース)

③M-14935 スイッチング AC アダプター(USB AC アダプター) Type-C オス 5.1V3.8A

④C-15002 HDMI ケーブル タイプ A オス-タイプ D オス(micro) 2m

⑤S-15181 Raspbian 書込済 microSD カード 32GB

【ソフトウェア】

OS : Raspbian buster 10.3

Raspberry Pi OS (32-bit) with desktop and recommended software

アプリ : Python3.7.3(Raspbian buster 10.3 に初めから入っています)

アプリ : Visual Studio Code1.32.0

設定内容

OS : Raspbian buster 10.3

Raspberry Pi OS (32-bit) with desktop and recommended software

Image with desktop and recommended software based on Debian Buster

Version: May 2020 Release date: 2020-05-27

次のリンクより OS イメージをダウンロードし、Etcher 等で Raspberry Pi 用の SD カードに書き込みます。

<https://www.raspberrypi.org/downloads/raspbian-pi-os/>

SD カードを Raspberry Pi 本体に差し込み、起動後、[Next] ボタンをクリック、

- Set country では、country を「Japan」にして、Language は「Japanese」、Timezone は「Tokyo」で[Next] ボタンをクリック、
- Change Password は、空欄のまま[Next] ボタンをクリック、
- Set Up Screen は、そのまま[Next] ボタンをクリック、
- Select WiFi Network は、[Skip] ボタンをクリック、
- Update Software も、[Skip] ボタンをクリック、
- Setup Complete では、[Restart] ボタンをクリックして再起動します。

日本語入力ができるように Raspbian の LXTerminal で次のコマンドをうちこみ、fcitx-mozc をインストールしました(要インターネット接続)。

```
$ sudo apt install fcitx-mozc
```

```
続行しますか？[Y/n]y
```

再起動(Reboot)後、[半角/全角]キーで日本語入力の切り替えができるようになります。

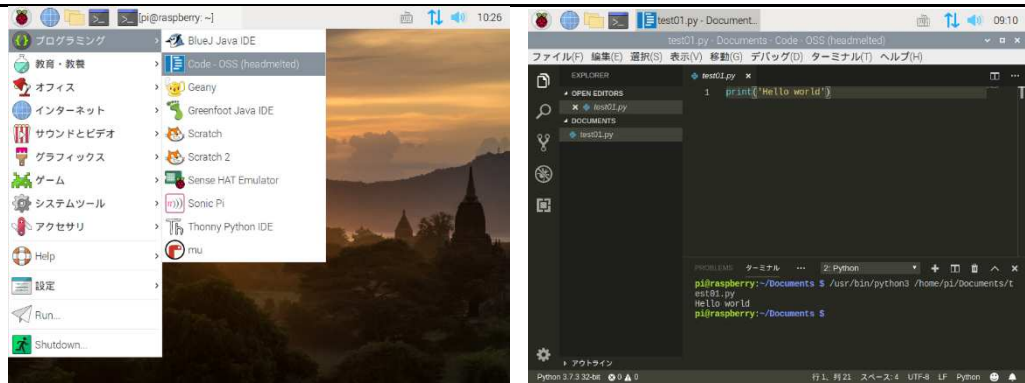
アプリ : Visual Studio Code1.32.0

エディタが使えるように Raspbian の LXTerminal で次のコマンドをうちこみ、Visual Studio Code をインストールしました(要インターネット接続)。

```
$ sudo -s
```

```
#. <( wget -O - https://code.headmelted.com/installers/apt.sh )
```

スタートメニューに プログラミング > Code-OSS(headmelted という名前で)登録されます。



Visual Studio Code の日本語化をするために、Code-OSS を起動し、メニュー **view > Extensions** をクリックします。
 入力欄に **python** と打ち込み、現れた Python の **[Install]** をクリックします。
 入力欄に **Japanese** と打ち込むと
 (Japanese Language Pack for Visual Studio Code が現れるので)
 Japanese Language Pack for Visual Studio Code の **[Install]** をクリック
 します。右下に出る青い **[Restart now]** ボタンをクリックすると再起動され、日本語
 化を確認できます。

Visual Studio Code から Python を実行するために、メニュー **表示 > コマンドパ
 レット** をクリックし、**select** と入力します。
 表示された中にある **[Python: インタープリター]** を選択し、「**Python 3.7.3 32-
 bit**」を選択します(Python2.7.16.32 を選択してはいけません)。これで Visual
 Studio Code で編集している Python プログラムは右クリックにより実行できるよ
 うになります。
 また、メニュー **ファイル > ユーザー設定 > 配色テーマ > monokai** にします。
 □起動時にウェルカムページを表示する のチェックを外します。
 右下に出てくる通知の **[×**] をクリックし、表示を消します。([Don't show again])

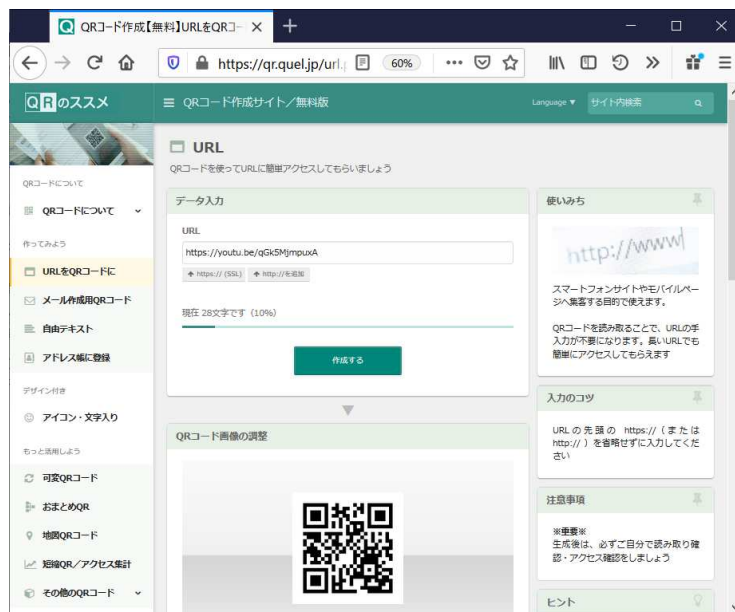
ファイル: **ファイルの入った「入力ミス探し」フォルダをデスクトップに保存しておきます。**

2 動画の登録

Windows10 の操作ビデオは AG-デスクトップレコーダーで、Raspberry Pi の操作ビデオは SimpleScreenRecorder で作成しました。イメージ動画は、Microsoft パワーポイントで作成しました。

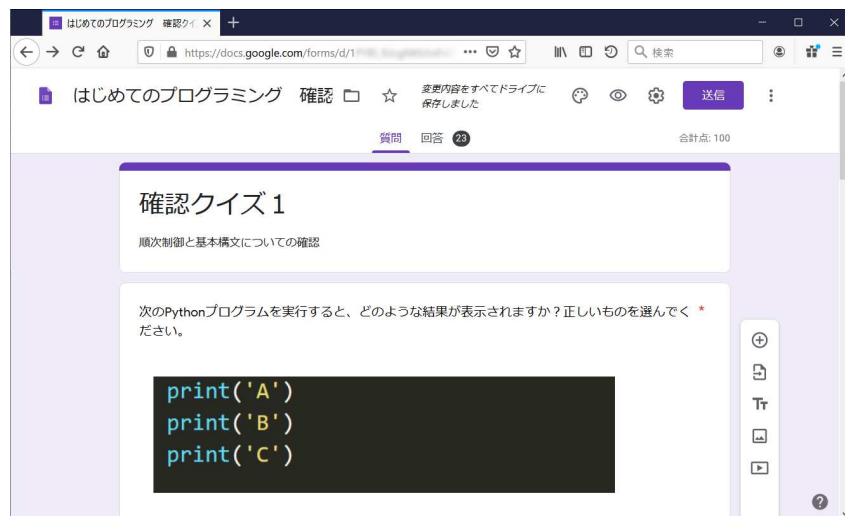
各動画は、youtube(<https://www.youtube.com/>)で「いいえ、子ども向けではありません」、「限定公開」として登録しています。

動画をスマホで見ることができるよう、テキストでは動画 URL を QR コードで示しています。QR コードは、Web ページ QR のスメ(<https://qr.quei.jp/url.php>)を活用して作成しました。



3 確認クイズの作成

確認クイズは、Web ページの Google フォーム(<https://docs.google.com/forms/u/1/>)で作成しました(要 Google アカウント)。



Google フォームで、プラスアイコン + をクリックします。

右上の設定アイコン 設定 をクリックします。

[テスト] 次に [テストにする] をクリックします。

省略可: メールアドレスを収集するには、[全般] 次に [メールアドレスを収集する] をクリックします。

[保存] をクリックします。

[送信] ボタンをクリックし、

送信方法 リンク を選択します。

URL を短縮にチェックを入れ、[コピー] ボタンをクリックします。

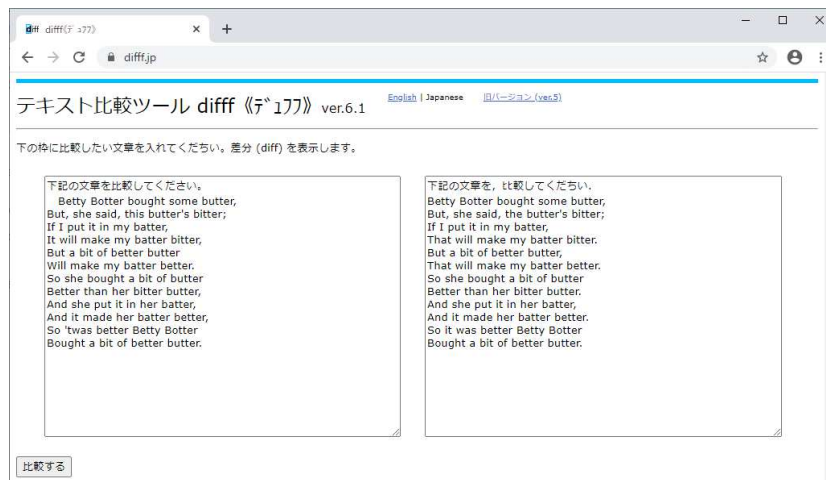
なお、Google フォームは、次のような設定にしました。メールアドレスなど個人を特定できる情報収集をしない設定です。

| 設定 | 設定 | 設定 |
|--|--|---|
| 全般 | 全般 | 全般 |
| <input type="checkbox"/> メールアドレスを収集する | <input type="checkbox"/> 進行状況バーを表示 | <input checked="" type="radio"/> テストにする 質問の点数を割り当てて自動採点を有効にできます。 |
| <input type="checkbox"/> 回答のコピーを送信 ① | <input type="checkbox"/> 質問の順序をシャッフルする | テストオプション |
| ログインが必要: | <input checked="" type="checkbox"/> 別の回答を送信するためのリンクを表示 | 成績の表示: |
| <input type="checkbox"/> 回答を 1 回に制限する | 確認メッセージ: 回答を記録しました。 | <input checked="" type="radio"/> 送信直後 |
| 回答者が行える操作: | | <input type="radio"/> 確認後に手動で表示する メール収集がオンになります |
| <input checked="" type="checkbox"/> 送信後に編集 | | 回答者が表示できる項目: |
| <input type="checkbox"/> 概要グラフとテキストの回答を表示 | | <input checked="" type="checkbox"/> 不正解だった質問 ② |
| | | <input checked="" type="checkbox"/> 正解 ② |
| | | <input checked="" type="checkbox"/> 点数 ② |

動画をスマホで見ることができるよう、動画 URL を QR コードで示しています。QR コードは、Web ページ QR のススメ(<https://qr.que1.jp/url.php>)を活用して作成しました。

4 テキスト比較サイトへのショートカットの作成

本教材では、テキスト比較サイト diff(<https://diff.jp/>)の持つ左の欄のテキストと右の欄のテキスト比較を活用して「入力ミス探し」機能を作成しました。答え合わせとしても使えるものです。



「入力ミス探し」機能は、テキスト比較サイト diff の左の欄にあらかじめ解答が入るようにして実現させています。テキスト比較サイト diff(<https://diff.jp/>)を活用したショートカットをプログラムの数(29)分作成しました。



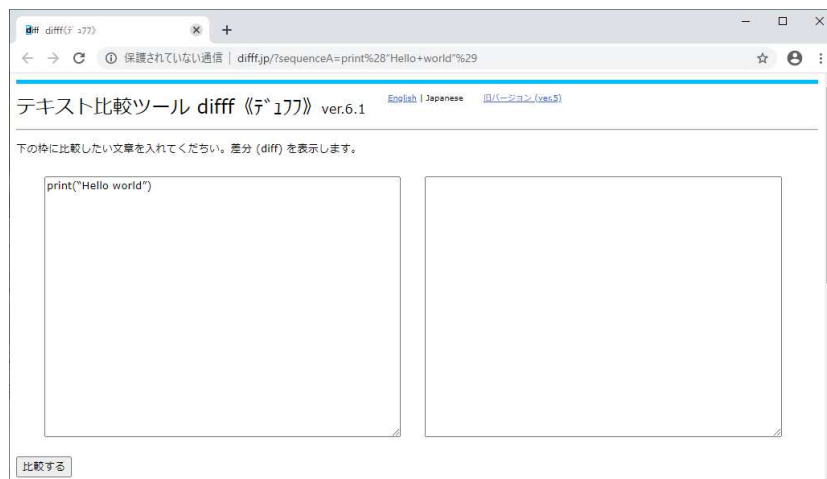
html の拡張子を持った 29 のファイルの内容は、Web ページへのリンクの記述となっています。次に示すように Web ページへのリンクを工夫し、テキスト比較サイト diff の左の欄に値が入るようにしています。

<http://diff.jp/?sequenceA=値>

たとえば、「test01 の入力ミス探し.html」ファイルの場合は、次のようになっています。

```
<!DOCTYPE html>
<script>window.location.href = "http://diff.jp/?sequenceA=print%28%E2%80%9CHello+world%E2%80%9D29"</script>
```

これにより、テキスト比較サイト diff の左の欄に「test01.py」のプログラムが入った状態になるわけです。



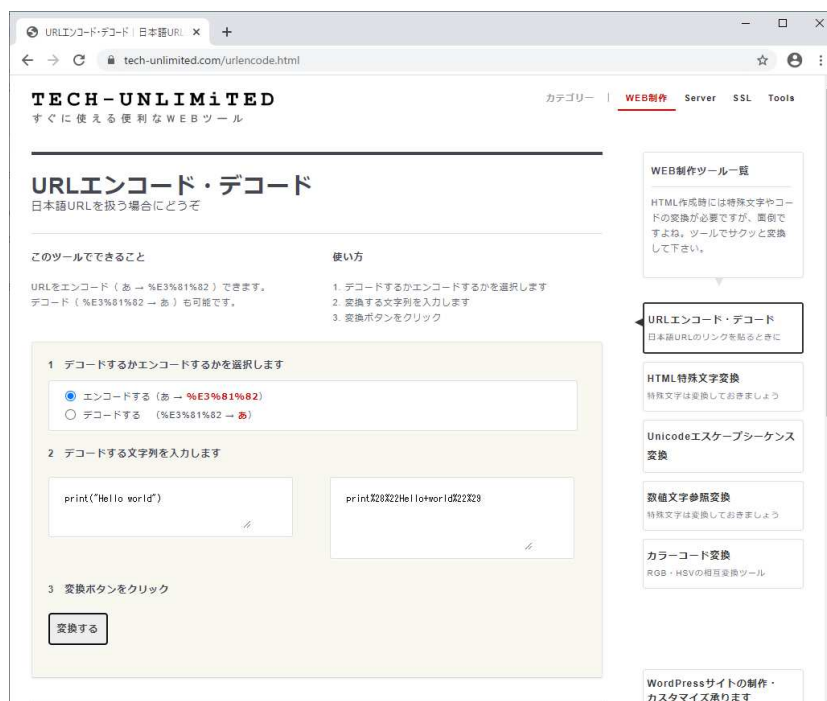
例に示した通り、左の欄に入れる値は、URL エンコードしたものにする必要があります。

たとえば、

`print("Hello world")` は、URL エンコードすると
`print%28%22Hello+world%22%29` になります。

URL エンコードできる Web サイト

TECH-UNLIMITED(<https://tech-unlimited.com/urlencode.html>)を活用してエンコードし、ショートカットを作成しました。



テキスト比較サイトへのショートカットの使い方は、「別冊_エラーと対処方法」で紹介しています。

5 Python でできる様々な処理のためのライブラリ・インストール

テキストの P.56 に「Python でできる様々な処理」として、3 つのプログラムを示しました。それぞれのようにしたら実行できるようになるのか説明します。

5-1 事務処理の自動化プログラムを実行するために

(例: ファイルの名前を変更するプログラム)

| | |
|---|--------------------------------------|
| 1 | import os |
| 2 | |
| 3 | path1 = './レポート.docx' # 変更前のファイル名 |
| 4 | path2 = './報告書.docx' # 変更後のファイル名 |
| 5 | os.rename(path1, path2) # ファイル名を変更する |

このプログラムは、A・B・C いずれのタイプからも特別なライブラリのインストールなく実行できます。

1 行目の `import os` により、`os` ライブラリを使うことができるようになります。`os` ライブラリは、Windows10 の Python3 にも WinPython にも、また Raspberry Pi の Raspbian OS の Python3 にも、初めからインストールされています。

`os` ライブラリを使うと、大量のファイル名の変更などの事務処理もプログラムで実施することができます。

5-2 機械の制御を実行するために

(例: Raspberry Pi で GPIO に接続した LED を点滅させるプログラム)

| | |
|----|--|
| 1 | import RPi.GPIO as GPIO |
| 2 | import time |
| 3 | |
| 4 | GPIO.setmode(GPIO.BCM) |
| 5 | GPIO.setup(15, GPIO.OUT) # LED を接続した GPIO15 を OUTPUT に設定 |
| 6 | while True: # ずっと繰り返す |
| 7 | GPIO.output(15, True) # GPIO15 (LED) を点灯 |
| 8 | time.sleep(2) # 2 秒待つ |
| 9 | GPIO.output(15, False) # GPIO15 (LED) を消灯 |
| 10 | time.sleep(2) # 2 秒待つ |
| 11 | GPIO.cleanup() |

C タイプ(Raspberry Pi)のみ実行できるプログラムです。

`RPi.GPIO` は、Python で Raspberry Pi の入出力ピンを制御するライブラリです。

プログラム実行前に大きく 2 つのことを実施する必要があります。

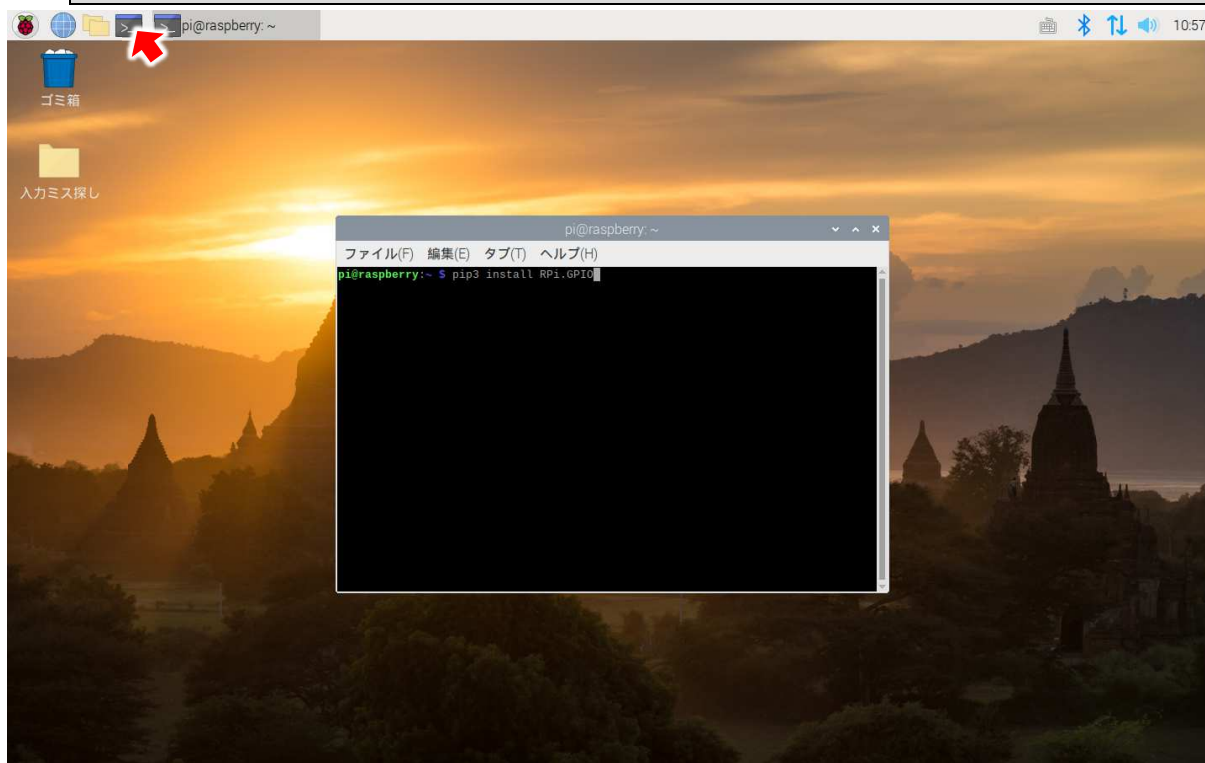
(1) `RPi.GPIO` ライブラリのインストール(要インターネット接続)

まず、`RPi.GPIO` ライブラリをインストールする必要があります。

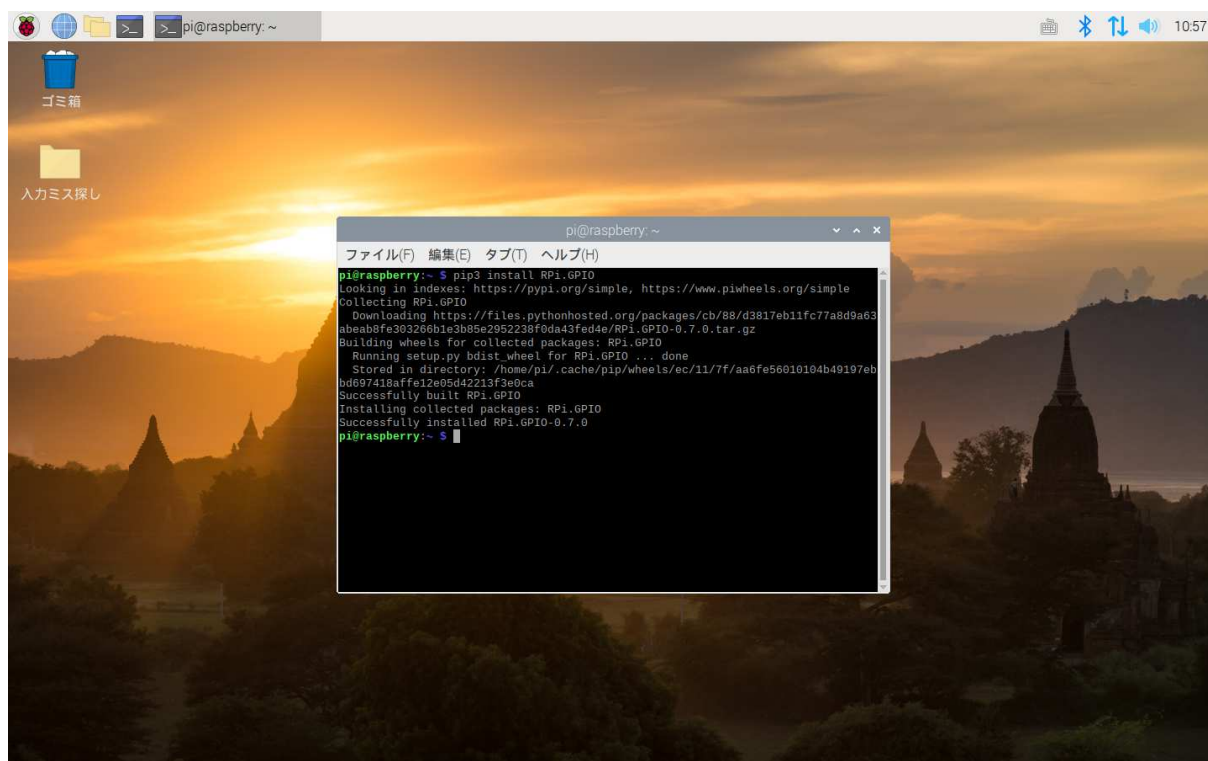
インストール方法は次の通りです。

- ① 赤い矢印で示した場所をクリックすると次の図のように「LXTerminal」が起動します。
- ② 「LXTerminal」で `pip3 install RPi.GPIO` と打ち込み[Enter]キーを押します。

```
$ pip3 install RPi.GPIO
```



- ③ これで RPi.GPIO ライブラリがインストールされ、Python3 から使えるようになります。なお、プログラムの1行目の `import RPi.GPIO as GPIO` は、RPi.GPIO ライブラリを(GPIO という名前で)使うことができるようにするという意味の記述です。



(2) Raspberry Pi に LED を接続する

次に Raspberry Pi の GPIO(General Purpose Input/Output)つまり汎用入出力に LED を接続する必要があります。

Raspberry Pi の USB ポートを下にしたとき GPIO は次のようなピン配置となっています。

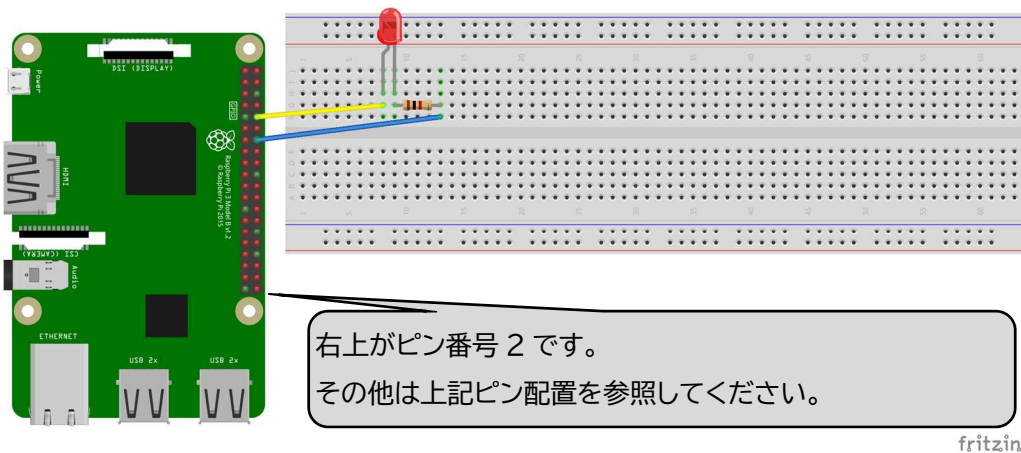
ピン配置

| 機能 | ピン番号 | ピン番号 | 機能 |
|---------|------|------|---------|
| 3.3V | 1 | 2 | 5V |
| GPIO 2 | 3 | 4 | 5V |
| GPIO 3 | 5 | 6 | GND |
| GPIO 4 | 7 | 8 | GPIO 14 |
| GND | 9 | 10 | GPIO 15 |
| GPIO 17 | 11 | 12 | GPIO 18 |
| GPIO 27 | 13 | 14 | GND |
| GPIO 22 | 15 | 16 | GPIO 23 |
| 3.3V | 17 | 18 | GPIO 24 |
| GPIO 10 | 19 | 20 | GND |
| GPIO 9 | 21 | 22 | GPIO 25 |
| GPIO 11 | 23 | 24 | GPIO 8 |
| GND | 25 | 26 | GPIO 7 |
| ID_SD | 27 | 28 | ID_SC |
| GPIO 5 | 29 | 30 | GND |
| GPIO 6 | 31 | 32 | GPIO 12 |
| GPIO 13 | 33 | 34 | GND |
| GPIO 19 | 35 | 36 | GPIO 16 |
| GPIO 26 | 37 | 38 | GPIO 20 |
| GND | 39 | 40 | GPIO 21 |

※ GPIO2、3 は抵抗付き GPIO です。

※ ID_SD と ID_SC は、外付けの EEPROM と I²C で接続するためのピンです。

プログラム例では、GPIO15を使うようにしていますので、GPIO15と GND の間に LED と 100Ωの抵抗を向きに気を付けながら直列接続します。



(画像) 接続図は、fritzing で作成しました、<https://fritzing.org/>

(1)と(2)ができていれば、Raspberry Pi から LED を点滅させるプログラムを実行させることができます。

LED を SSR(半導体リレー)にすれば、大型機械の制御もプログラムで実施することができます。

5-3 画像認識を実行するために

(例: カメラの画像を表示させるプログラム)

| | | |
|----|--|-----------------|
| 1 | <code>import cv2</code> | |
| 2 | | |
| 3 | <code>cap = cv2.VideoCapture(0)</code> | # 0 はカメラのデバイス番号 |
| 4 | <code>while True:</code> | # ずっと繰り返す |
| 5 | <code> ret, frame = cap.read()</code> | # カメラから画像を読み込む |
| 6 | <code> cv2.imshow('camera capture', frame)</code> | # 画像をウィンドウに表示する |
| 7 | <code> k = cv2.waitKey(1)</code> | # キー入力を 1m 秒待つ |
| 8 | <code> if k == 27:</code> | # [ESC]キーを押したら、 |
| 9 | <code> break</code> | # 繰り返しから抜ける |
| 10 | <code>cap.release()</code> | |
| 11 | <code>cv2.destroyAllWindows()</code> | # ウィンドウを消す |

openCV ライブラリのインストールが必要なプログラムです。

cv2 とは、Python で画像処理・画像解析等をするライブラリ openCV(Open Source Computer Vision Library)のことです。

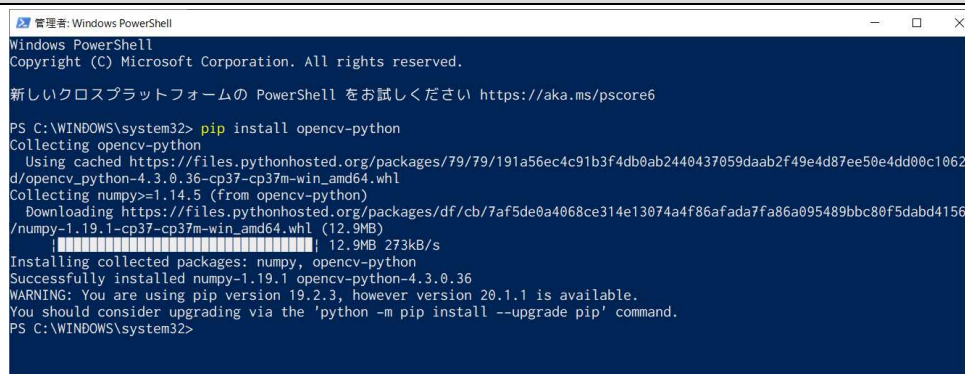
このプログラムは、USB 接続した Web カメラの画像を[ESC]キーが押されるまで表示し続けるものです。Web カメラが搭載されていないコンピュータには、Web カメラを USB 端子に接続する必要があります。

- A タイプのパソコンでのインストール方法(要インターネット接続)

Windows PowerShell を管理者として実行したうえで

`pip install opencv-python` と打ち込み[Enter]キーを押します。

```
> pip install opencv-python
```



```
管理: Windows PowerShell
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

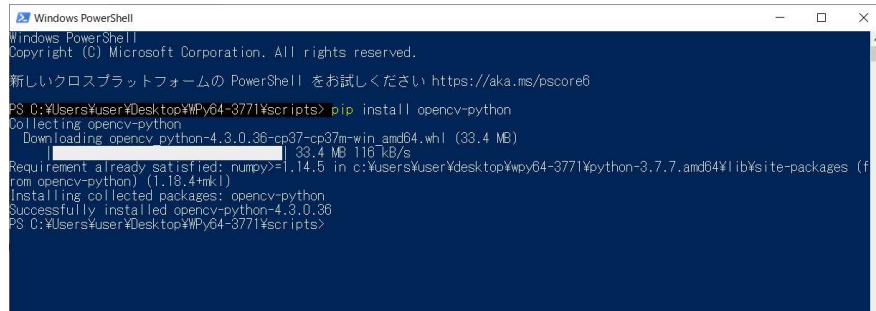
PS C:\WINDOWS\system32> pip install opencv-python
Collecting opencv-python
  Using cached https://files.pythonhosted.org/packages/79/79/191a56ec4c91b3f4db0ab2440437059daab2f49e4d87ee50e4dd00c1062d/opencv_python-4.3.0.36-cp37-cp37m-win_amd64.whl
Collecting numpy>=1.14.5 (from opencv-python)
  Downloading https://files.pythonhosted.org/packages/df/cb/7af5de0a4068ce314e13074a4f86afada7fa86a095489bbc80f5dabd4156/numpy-1.19.1-cp37-cp37m-win_amd64.whl (12.9MB)
    |#####| 12.9MB 273kB/s
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.19.1 opencv-python-4.3.0.36
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.
PS C:\WINDOWS\system32>
```

※ なお、このコマンドにより、openCV の最新版がインストールされます。

これで openCV ライブラリがインストールされ、Python3 から使えるようになります。プログラムの 1 行目の `import cv2` により openCV ライブラリを使うことができるようになります。

- Bタイプのパソコン WinPython 環境でのインストール方法(要インターネット接続)
 デスクトップにある「WPy64-3771」フォルダの中にある
 WinPython Powershell Prompt.exe をダブルクリックしたうえで
`pip install opencv-python` と打ち込み[Enter]キーを押します。

> `pip install opencv-python` ↵



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

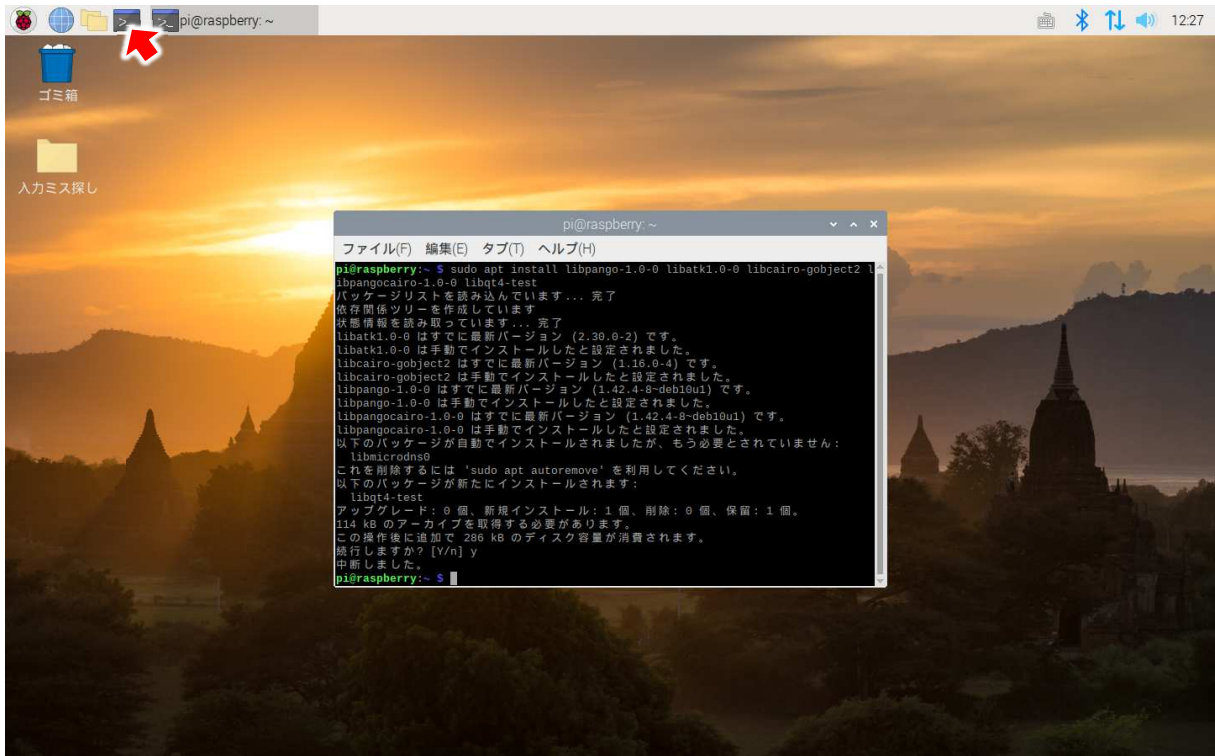
新しいクロスプラットフォームの PowerShell をお試しください https://aka.ms/pscore6

PS C:\Users\User\Desktop\WPy64-3771\Scripts> pip install opencv-python
Collecting opencv-python
  Downloading opencv-python-4.3.0.36-cp37-cp37m-win_amd64.whl (33.4 MB)
    |#####| 33.4 MB 116 kB/s
Requirement already satisfied: numpy>=1.14.5 in c:\users\user\desktop\wpy64-3771\python-3.7.7.amd64\lib\site-packages (from opencv-python) (1.18.4+mk1)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.3.0.36
PS C:\Users\User\Desktop\WPy64-3771\Scripts>
  
```

※ なお、このコマンドにより、openCV の最新版がインストールされます。

これで openCV ライブラリがインストールされ、Python3 から使えるようになります。プログラムの 1 行目の `import cv2` により openCV ライブラリを使うことができるようになります。

- Cタイプの Raspberry Pi でのインストール方法(要インターネット接続)
 ① 赤い矢印で示した場所をクリックすると次の図のように「LXTerminal」が起動します。

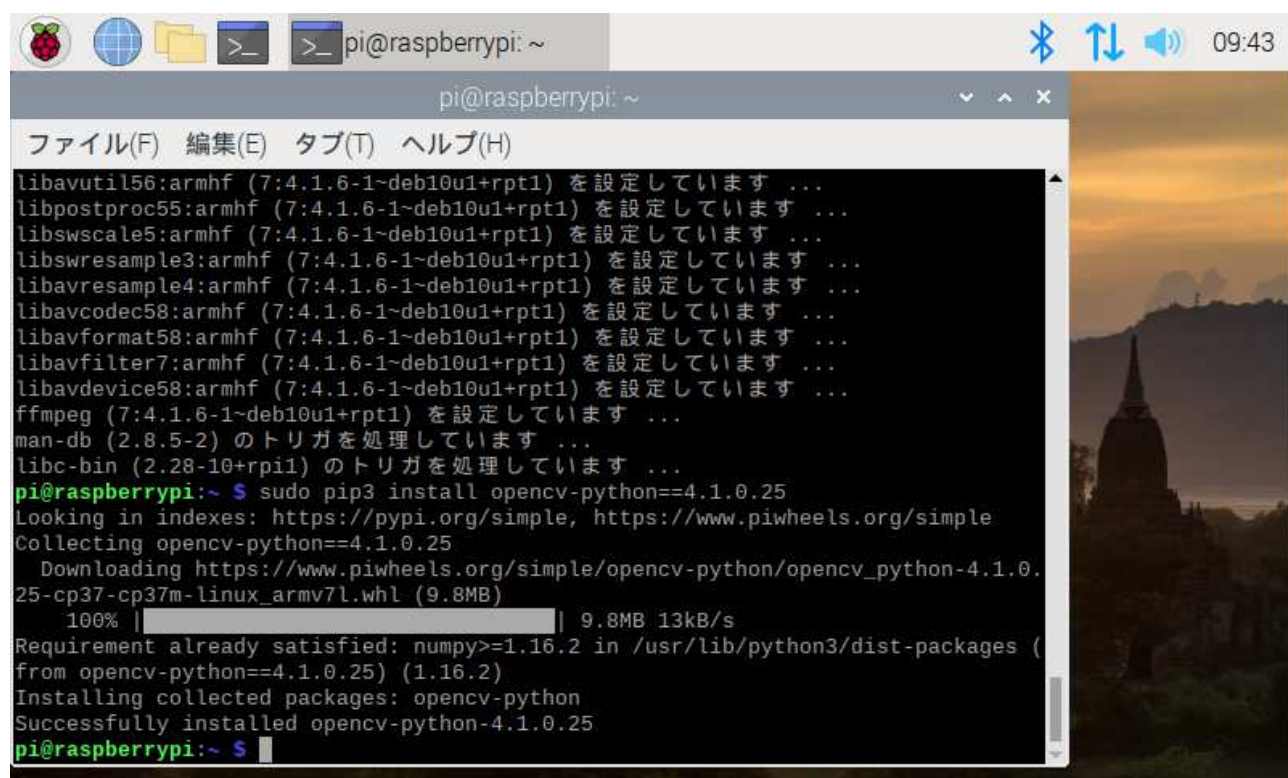


② 次のコマンドを打ち込み[Enter]キーを押します。

```
$ sudo apt install libpango-1.0-0 libatk1.0-0 libcairo-gobject2 libpangocairo-1.0-0 libqt4-test
続行しますか？ [Y/n]y
$ sudo apt install libtiff5 libqtcore4 libwebp6 libavcodec58 libavutil56 libqtgui4 libavformat58 libgdk-pixbuf2.0-0
続行しますか？ [Y/n]y
$ sudo pip3 install opencv-python==4.1.0.25

$ sudo apt install libatlas-base-dev
$ sudo apt install libjasper-dev
$ sudo apt install qt4-dev-tools qt4-doc qt4-qtconfig libqt4-test
```

sudo はスーパーユーザー権限で実行する命令です。パスワードを尋ねてきますので入力してください。また、初回、インストールしてよいか[Y/n](Yes か No か)を尋ねてきますので、[y]キーを押し[Enter]キーを押すようにしてください。



```
pi@raspberrypi: ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
libavutil56:armhf (7:4.1.6-1~deb10u1+rpt1) を設定しています ...
libpostproc55:armhf (7:4.1.6-1~deb10u1+rpt1) を設定しています ...
libswscale5:armhf (7:4.1.6-1~deb10u1+rpt1) を設定しています ...
libswresample3:armhf (7:4.1.6-1~deb10u1+rpt1) を設定しています ...
libavresample4:armhf (7:4.1.6-1~deb10u1+rpt1) を設定しています ...
libavcodec58:armhf (7:4.1.6-1~deb10u1+rpt1) を設定しています ...
libavformat58:armhf (7:4.1.6-1~deb10u1+rpt1) を設定しています ...
libavfilter7:armhf (7:4.1.6-1~deb10u1+rpt1) を設定しています ...
libavdevice58:armhf (7:4.1.6-1~deb10u1+rpt1) を設定しています ...
ffmpeg (7:4.1.6-1~deb10u1+rpt1) を設定しています ...
man-db (2.8.5-2) のトリガを処理しています ...
libc-bin (2.28-10+rpil) のトリガを処理しています ...
pi@raspberrypi:~$ sudo pip3 install opencv-python==4.1.0.25
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting opencv-python==4.1.0.25
  Downloading https://www.piwheels.org/simple/opencv-python/opencv_python-4.1.0.25-cp37-cp37m-linux_armv7l.whl (9.8MB)
    100% |#####| 9.8MB 13kB/s
Requirement already satisfied: numpy>=1.16.2 in /usr/lib/python3/dist-packages (from opencv-python==4.1.0.25) (1.16.2)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.1.0.25
pi@raspberrypi:~$
```

これで openCV ライブラリがインストールされ、Python3 から使えるようになります。プログラムの 1 行目の `import cv2` により openCV ライブラリを使うことができます。

openCV ライブラリに用意されているたくさんの命令を活用すれば、画像処理や画像認識もプログラムで実施することができます。