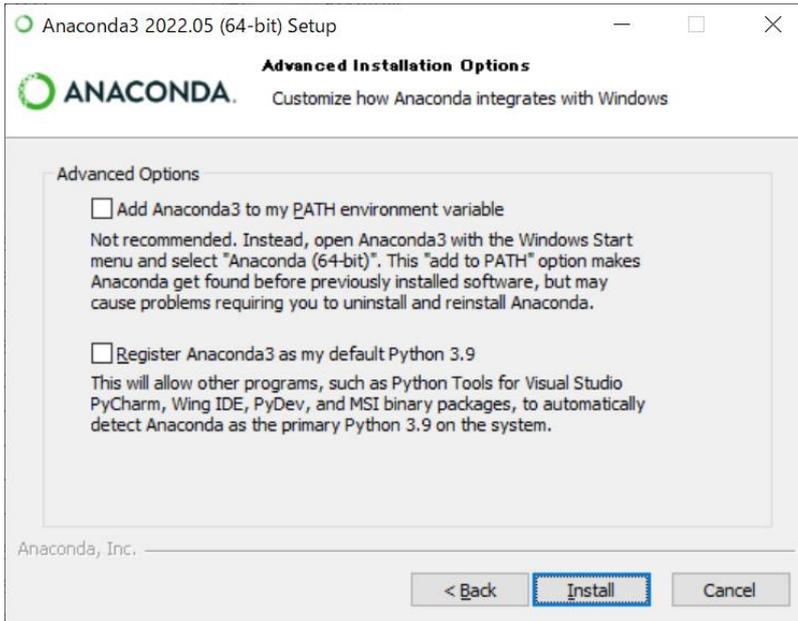


ドローンプログラムの作成

準備①

- <https://www.anaconda.com/distribution/#download-section>より、Anaconda をダウンロードしてインストール。設定は、Advanced Option でどちらもチェックが外れていることを確認する。
その他の設定は全てデフォルト。



スタートメニューから「Anaconda Prompt」を検索し起動する。コマンドライン(黒い画面に文字のみ)が表示される。

以下のコマンドを入力して、必要なライブラリをダウンロードする。

1. 今回使用するドローン用の仮想環境「drone1」を作成。仮想環境で作業を行う。

```
conda create -n drone1
activate drone1
```

仮想環境が実行されているとき、(drone1)が表示される。されていない場合は再度上記の activate コマンドを実行すること。

```
C:¥Users¥m.tamaoka>activate drone1
(drone1) C:¥Users¥m.tamaoka>
```

2. ドローンの自動操縦に必要なライブラリをダウンロード。

```
conda install pip
pip install numpy
pip install djitellopy
```

ドローンプログラムの作成

以下のような表示が出て停止したら、Enter を押して続行する。

```
Proceed ([y]/n)?
```

- サンプル・プログラムを動かしてみる。以下のコマンドを入力する。
 1. サンプル・プログラムをダウンロードする

```
conda install git
git clone https://github.com/damiafuentes/DJITelloPy.git
```
 2. サンプル・プログラムのあるフォルダに移動する。サンプル・プログラム(take-picture)を実行する。※この時点ではエラーが出て実行できない。

```
cd DJITelloPy/examples
python take-picture.py
```
- ドローンと PC を接続する。接続は Wi-Fi を用いて行う。

まず、ドローンにバッテリーを入れ、電源を ON にする。Tello 自体がアクセス・ポイントとなり、PC やスマートフォンから接続できるようになる。

PC の Wi-Fi アクセス・ポイント一覧から、指定のドローン名を選択する。

以下のページからファイヤーウォールを無効にする。
<https://www.buffalo.jp/support/faq/detail/16417.html>

設定が完了したら、以下のコマンドを再実行する。キーボードの上キーを押せば、コマンドの履歴から再実行できる。

```
python take-picture.py
```

ドローンが離陸～着陸すれば OK。

サンプル・プログラムの内容を確認する。以下のコマンドで jupyter-notebook を立ち上げる。

```
conda install jupyter
jupyter notebook
```

ブラウザに、examples フォルダの内容が表示される。その中から、先ほど実行した「take-picture.py」を選択して開く。コードの内容は次のページの通り。

ドローンプログラムの作成

```
import cv2 # 画像処理のライブラリ
from djitellopy import Tello # ドローン操作のライブラリ

tello = Tello() # ドローン接続の準備
tello.connect() # ドローンと接続する

tello.streamon() # カメラの起動
frame_read = tello.get_frame_read() # カメラから映像を取得する

tello.takeoff() # ドローンを離陸させる
# カメラから取得した映像を画像として保存
cv2.imwrite("picture.png", frame_read.frame)

tello.land() # ドローンを着陸させる
```

examples フォルダ内に、picture.png という画像ファイルがある。開くと、ドローンで撮影した写真が確認できる。

ドローンプログラムの作成

準備②

- ドローン飛行シミュレーターを使えるようにする。
今回は、簡易的なシミュレーターを使用して開発する。
Anaconda prompt を再起動して、ルートフォルダに戻る。以下のコマンドを実行する。
`explorer .`
ファイルエクスプローラーが起動する。配布フォルダから「tello_sim.zip」をダウンロードして解凍し、表示されたフォルダ内へコピーする。
- Anaconda prompt に戻り、以下のコマンドを実行する
`activate drone1`
`cd tello_sim`
`pip install .`
`jupyter notebook`
jupyter notebook が起動し、ブラウザに表示される。
「example_flight_notebook.ipynb」をクリックして開く。実行したいセルを選択して、上のメニューの[Run]をクリックするとプログラムが実行され、実行結果のグラフが表示される。

```
import tello_sim as Tello # シミュレータライブラリを参照する
tello = Tello.Simulator() # シミュレーターの準備

tello.takeoff() # ドローンを離陸させる

tello.move_forward(100) # 前方へ 100cm 進む
tello.move_left(100) # 左へ 100cm 進む
tello.flip("f") # 前方に宙返りする
tello.rotate_counter_clockwise(180) # 反時計回りに 180 度回転する
tello.move_forward(40) # 前方に 40cm 進む
tello.move_up(25) # 上に 25cm 上昇する

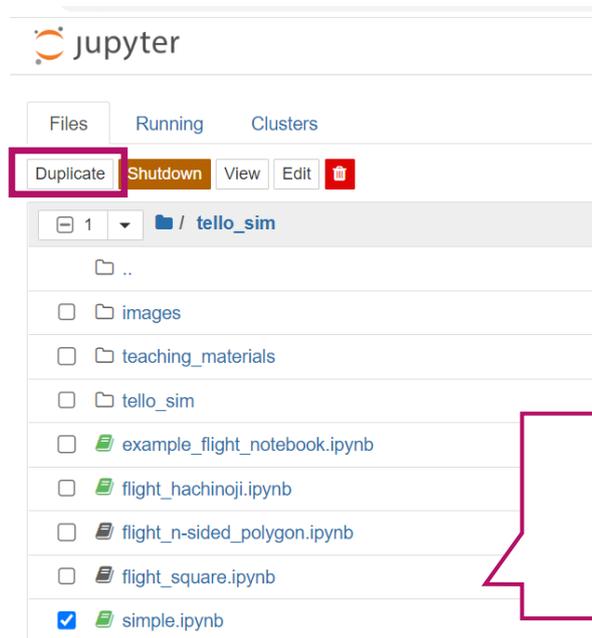
tello.land() # ドローンを着地させる
```

ドローンプログラムの作成

課題

- 以下のプログラムを作成する。
 - ① 離陸地点を一つの角として 1 辺 50cm の正方形を描き、最初の地点へ着陸する
 - ② 離陸地点を中心として 8 の字を描き、最初の地点へ着陸する
 - (ア) 円の軌道は正八角形を代わりに用いる
 - (イ) 8 角形の 1 辺は 30cm とする
 - (ウ) 繰り返しの処理は for 文を用いる
 - ③ 数値を入力させる。入力した数の正多角形を描くプログラムを作成する
 - (ア) 正多角形の 1 辺は 40cm とする
 - (イ) 入力できる数値は 3 以上 10 以下とする
 - (ウ) 繰り返しの処理は for 文を用いる
- 課題の進め方は以下の通り
 - ① シミュレーターでプログラムを作成する。

Jupyter notebook から「tello_sim」フォルダを開き、「simple.ipynb」を複製 (Duplicate) する。課題ごとに複製したファイルを作成し編集する。



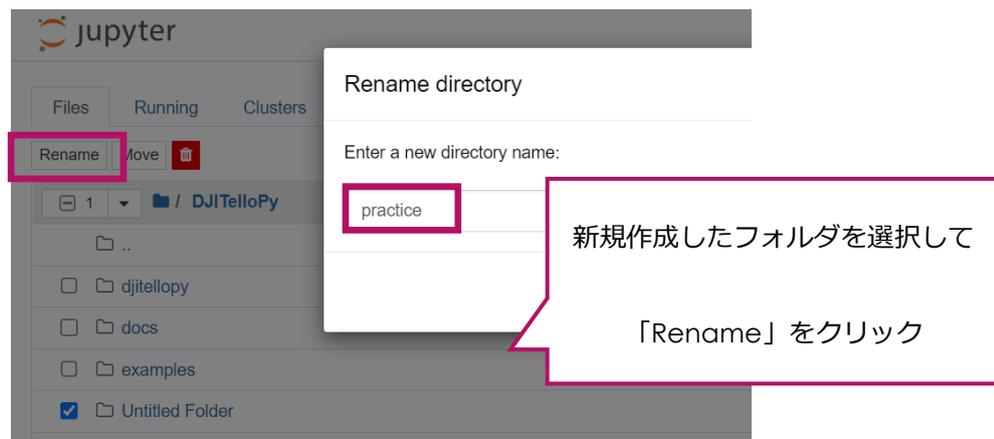
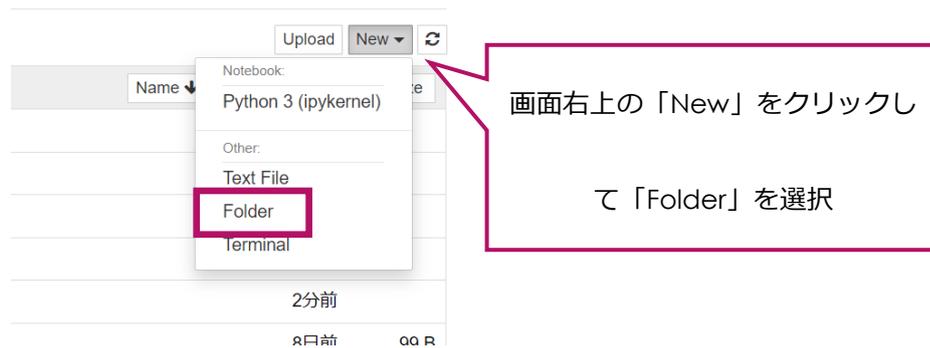
simple.ipynb を選択して

「Duplicate」をクリック

ドローンプログラムの作成

- ② シミュレーターでの実行結果が問題ないことを確認した後は、実機のドローンを実際に飛行させる。

Jupyter notebook から「DJITelloPy」フォルダを開き、フォルダ「practice」を作成する。「practice」内に課題ごとに Python ファイルを作成してプログラムを実装する。着陸から着地までをコピーすれば実装できるようになっている。



```
from djitellopy import Tello

tello = Tello()

tello.connect()
tello.takeoff()

# 編集箇所ここから

# 編集箇所ここまで

tello.land()
```

作成する Python ファイルの例

ドローンプログラムの作成

- ③ 実装が終わったら、anaconda prompt を起動し、以下のコマンドを実行する。
プログラムの実行前に Tello を起動し、PC につなげる。※ファイル名例：

practice01.py

```
cd DJITelloPy/practice
python practice01.py
```

- ④ 仕様通りにプログラムが動くことを確認したら、課題ファイル(.py)①②③を提出フォルダに自分の名前のフォルダを作成し提出する。

フォルダ構成

[ルートディレクトリ] # anaconda prompt を実行した直後のフォルダ

```
- [ DJITelloPy]
    - [examples]
        simple.py
        take-picture.py
    - [practice]
        practice01.py
        practice02.py
        practice03.py
- tello_sim
    example_flight_notebook.ipynb
    example
```