



IoT プログラミング講座 (1)

益田プログラミング研究会

Masuda Programming Research Society (MPRS)

<https://maro-v.jp/~nagata-iics/>

永田 武



#1 Linux 入門、はじめてのプログラミング

(1) 本日の内容 01_iot

1. Linux 入門

(1) はじめに
標準入力と標準出力

(2) キーボードに慣れよう

```
python abc.py
python abc2.py
python abc3.py
```

2 Linux 入門

- (1) ifconfig (IP アドレス)
- (2) cat /etc/os-release (OS のバージョン)
- (3) python -V (Python のバージョン)
- (4) date (現在の日時)
- (5) cal (カレンダー)
- (6) ls (ファイルの一覧)
- (7) xeyes
- (8) xcalc
- (9) pwd (現在のディレクトリ)
- (10) cd (ホームディレクトリに移動)
cd 2025 (2025 ディレクトリに移動)
cd iot01 (iot01 ディレクトリに移動)
ls (ファイル一覧)
- (11) tree (木構造でファイル一覧)
- (12) mkdir iot01 (iot01 ディレクトリ作成)
mkdir iot02
mkdir iot03
mkdir iot04
mkdir iot05
mkdir iot06
mkdir iot07
- (13) man ls (オンラインマニュアルで ls コマンド調べる)
- (14) history (実行コマンドの履歴)

3 はじめてのプログラミング

(1) test.py の作成と動作確認
gedit test.py & エディターの立ち上げ
(& はバックグラウンドで実行 (コマンドプロンプトが戻ってくる))

下記 (2 行) を入力して、保存 (Save)

```
# test.py
print('Hello world Python!')
```

実行 python test.py

- (2) プログラム開発の流れ
ソースファイルの作成
→ 実行 (エラーがあればソースファイルを修正)
- (3) プログラムの基本形 ([A]の部分は省略可能)

```
# ソースファイル名 (#はコメント)
import モジュール名

...
from モジュール名 import オブジェクト名 (関数、変数、
クラス)

...
def 関数名 (引数の列):
    return 戻り値の列

def main(): [A]
    手続きの記述

if __name__ == '__main__': [A]
    main() [A]
(一般に、プログラムは main から始まる)
```

(4) プログラムの改造
test.py を test2.py として修正

- コピー cp test.py test2.py
- エディタの起動 g test2.py&

(注) 今後は gedit の代わりに g を用いる
alias で調べられる

- 下記 (5 行) を入力して、保存 (Save)

```
# test2.py
def main():
    print('Hello world Python!')
if __name__ == '__main__': [A]
    main()
```

- 実行 p test2.py
(注) 今後は python の代わりに p を用いる

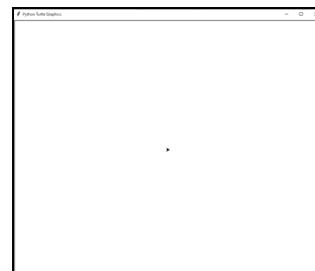
4. Turtle プログラミング

(1) 対話型インタプリタの利用

```
p
...
>>> ここにコマンド (命令) を書く
```

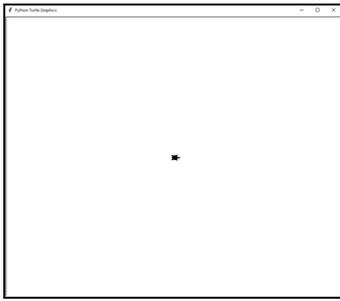
(1.1) タートルモジュールの読み込み

```
>>> from turtle import *
>>> t = Turtle()
```



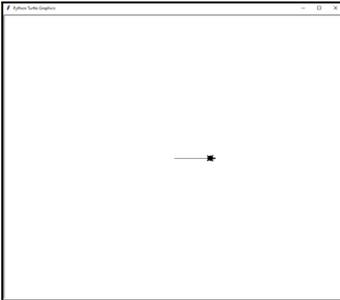
(1.2) タートルの形を変更

```
>>> t.shape('turtle')
```



(1.3) 前進 100 ステップ

```
>>> t.fd(100)
```



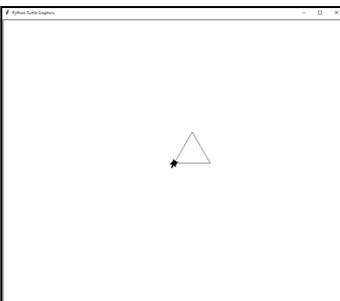
(1.4) 方向転換と前進

```
>>> t.lt(120)
```

```
>>> t.fd(100)
```

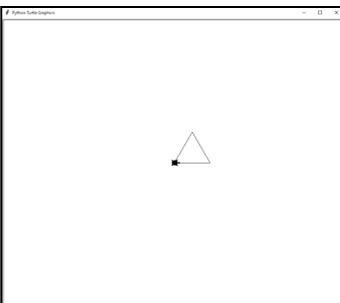
```
>>> t.lt(120)
```

```
>>> t.fd(100)
```



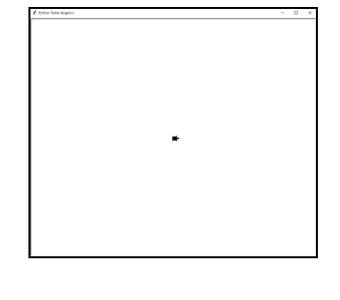
(1.5) 元の位置・方向に戻る

```
>>> t.home()
```



(1.6) 描いた図形をリアする

```
>>> t.clear()
```



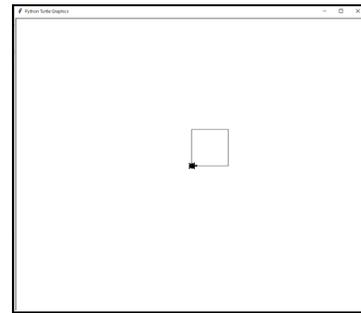
(1.7) 四角形を描いてみる

```
>>> t.fd(100);t.lt(90)
```

```
>>> t.fd(100);t.lt(90)
```

```
>>> t.fd(100);t.lt(90)
```

```
>>> t.fd(100);t.lt(90)
```



(1.8) 対話型インタプリタの終了

```
>>> exit()
```

おまけ

リダイレクション・パイプ・音声出力

```
cd jtalk
```

```
p jt1.py
```

```
p jt2.py
```

```
p jt3.py
```

```
p tenki.py
```

```
p tenki.py > in.txt | p jt.py
```

電源 OFF

```
poweroff
```

参考資料

(1) Python によるアルゴリズムとデータ構造の基礎

<https://www.coronasha.co.jp/np/isbn/9784339029079/>

(2) Raspberry Pi による IoT システム開発実習

<https://www.morikita.co.jp/books/mid/085571>

連絡先

t.nagata.wp@gmail.com 永田 武

<http://www.maro-v.jp/~nagata-iics/>