第 3 版 2021/11/11
山村 一繁

python を使うために（その１）

★「はじめのいーっぽ」からもう一歩進む

- 変数の値を 1 大きくする。
  - 「=」は代入を表わす。

```
a = 1
a = a + 1
print(a)
```

  - 「a += 1」としても同じ。
- 変数に文字を代入することもできる。

```
char1 = 'abc'
char2 = 'def'
char = char1 + char2
print(char)
```

  - print の中で使えば，どんな数値か示すことができる。

```
print('解 1 = ', ans1, ',  解 2 = ', ans2)
```

- 変数には日本語も使える。

```
りんご = '100円'
みかん = '50円'
print(りんご, みかん)
```

- ライブラリを使う。
  - 「from math import *」は，ひとつの例。
  - 「import math」・・・標準形。sin は math.sin と書く。
  - 「import math as mt」・・・math を mt と省略する。sin は mt.sin と書く。
  - ライブラリを使った例。二次方程式を解く。

```
import numpy
numpy.roots([1, -1, -2])
```

★ プログラムを作るための最低限の文法

- **if 文**
  - ある条件を満す場合にのみ実行する。
  - 「if 条件 :」という形で使う。
  - 実行する行は，字下げする。
  - 字下げは基本的に空白 4 つ。
  - Spyder などでは自動的に字下げしてくれる。
  - 値が等しいか判断するには「a == b」とする。

プログラム 1: src00.py

```
#!/usr/bin/env python3

# 解の公式を使って二次方程式を解く。
# Coded by YAMAMURA,K.
# 2019/11/03

# ライブラリのインポート
from math import *

# 係数の設定
a = 4
b = 4
c = 1

# 解の公式
ans1 = (-b + sqrt(b**2-4*a*c)) / (2*a)
ans2 = (-b - sqrt(b**2-4*a*c)) / (2*a)

# 文字変数の設定
label1 = '解１='
label2 = ',  解２='

# 解の出力
print(label1, ans1, label2, ans2)
```

プログラム 2: if01.py

```
#!/usr/bin/env python3

# 解の公式を使って二次方程式を解く。
# if 文を使ってみる。
# Coded by YAMAMURA,K.
# 2019/11/03

# ライブラリのインポート
from math import *

# 係数の設定
a = 4
b = 4
c = 1

# 判別式の計算
D = b**2 - 4*a*c
```

```
18
19 # 文字変数の設定
20 label1 = '解１='
21 label2 = ',  解２='
22
23 # if 文を使った出力
24 if D < 0:
25     print('判別式が負です')
26 else:
27     ans1 = (-b + sqrt(b**2-4*a*c)) / (2*a)
28     ans2 = (-b - sqrt(b**2-4*a*c)) / (2*a)
29     print(label1, ans1, label2, ans2)
```

プログラム 3: if02.py

```
1  #!/usr/bin/env python3
2
3  # 西暦から元号を調べる。
4  # if ... elif ... elif ... else を使う。
5  # Coded by YAMAMURA,K.
6  # 2019/11/03
7
8  # 西暦の設定
9  by = 2000
10
11 # 元号の出力
12 if by > 2018:
13     print('令和生れ')
14 elif by > 1988:
15     print('平成生れ')
16 elif by > 1925:
17     print('昭和生れ')
18 else:
19     print('大正以前の生れ')
```

- **繰り返し**
  - 同じ処理を繰り返す。
  - 「for 変数 in 繰り返し内容:」という形で使う。
  - 実行する行は，字下げする。
  - 「range(10)」は 0 から 10 個の整数。
  - 0 から 10 個の整数を足し合せる例。

プログラム 4: loop01.py

```
1  #!/usr/bin/env python3
2
3  # 1〜9 の和を計算する。
4  # for 文を使って繰り返し計算をする。
5  # Coded by YAMAMURA,K.
6  # 2019/11/03
7
8  # 合計の初期値の設定
9  sum = 0
10
11 # for 文を使って,和を繰り返す。
12 for i in range(10):
13     sum = sum + i
```

```
14      print('i = ', i, '小計 =',sum)
15
16  # 結果を出力する。
17  print('合計は, ', sum, 'です。')
```

- **リスト**
  - ひとつの変数に複数の値を設定する。

プログラム 5: list00.py

```
1   #!/usr/bin/env python3
2
3   # リストを使ってみる。
4   # matplotlib を使って,グラフで表わす。
5   # Coded by YAMAMURA,K.
6   # 2019/11/03
7
8   # グラフの日本語フォントの設定
9   igfont = {'family':'MS Gothic'}
10
11  # ライブラリのインポート
12  import matplotlib.pyplot as plt
13
14  # リストの設定
15  tokyo_tmps = [ 15.1, 15.4, 15.2, 15.4, 17.0, 16.0 ]
16
17  # リストをプロットする。
18  plt.plot(tokyo_tmps)
19  # x,y 軸のラベルの設定
20  plt.ylabel('東京の気温（℃）', **igfont)
21  plt.xlabel('日', **igfont)
22
23  # 表示する。
24  plt.show()
```

  - 「tokyo_tmps[0]」は最初の値を示し,「tokyo_tmps[1]」は二番目の値を示す。
  - 「max(tokyo_tmps)」・・・tokyo_tmps の中の最大の値
  - 「min(tokyo_tmps)」・・・最小の値。
  - 「sum(tokyo_tmps)」・・・合計。
  - 「len(tokyo_tmps)」・・・要素の数。
  - 「tokyo_tmps.sort()」・・・要素を小さい順に並べかえる。

※ このプログラムはオンライン実行環境ではなく，パソコンの Spyder，Jupyter Notebook などで実行すること。

- **関数**
  - プログラムの中に同じ処理がある場合に，その処理をまとめることができる。
  - 「def 関数名 (引数):」という形で使う。
  - 実行する行は，字下げする。

プログラム 6: func00.py

```
1   #!/usr/bin/env python3
2
3   # 解の公式を使って二次方程式を解く。
4   # 判別式の計算,解の計算に関数を使う。
5   # Coded by YAMAMURA,K.
```

```
# 2019/11/03

# ライブラリのインポート
from math import *

# 判別式を計算する関数
def funcD(b, c):
    return b**2 - 4*a*c

# 解を計算する関数
def funcANS(a,b,c):
    ans1 = (-b + sqrt(b**2-4*a*c)) / (2*a)
    ans2 = (-b - sqrt(b**2-4*a*c)) / (2*a)
    return ans1, ans2

# ------------------------------------------

# 係数の設定
a = 1
b = 5
c = 2

# 判別式の計算
D = funcD(b, c)

# if 文を使った出力
if D < 0:
    print('判別式が負です')
else:
    (ans1, ans2) = funcANS(a, b, c)
    print('解１=', ans1, ', 解２=', ans2)
```

- **ファイルの読み書き。**
  - 「ファイルオブジェクト名 = open(' ファイル名', 'r')」という形で使う。

プログラム 7: file01.py

```
#!/usr/bin/env python3

# 各班の模型の諸元から,断面二次モーメントを計算する。
# 模型のデータはタブ (\t)で区切られている。
# open 文を使って,ファイルを読み込む。
# Coded by YAMAMURA,K.
# 2019/11/03

# データファイルを開く
f = open('model2018.dat', 'r', encoding='utf-8')

# ファイルを一行づつ処理する
for line in f:
    # 行頭が # でなく,かつ空行でもない場合に計算する。
    if line[0] != '#' and len(line) != 1:
        # タブ (\t)を目印にデータをリストにする
        cols = line.split('\t')
        model = cols[0] # 先頭はモデルの名前
        D = float(cols[1]) # 二番目は D。文字列を実数に変換。
        b = float(cols[2]) # 二番目は d。文字列を実数に変換。
```

```
        I = b * D**3 / 12 # 断面二次モーメント I の計算
        print(model, ': b =', b, ', D = ', D, ' : I = ', I)

# データファイルを閉じる
f.close()
```

データ 1: model2018.dat

```
# 2018

#     おもりの数　推定値
# 厚さ 幅 長さ 厚 薄 E 振動数

a-1 3 30 100 0 1 1.5 5.325
a-2 3 30 100 1 0 1.5 4.297

b-1 3 20 150 0 0 1.0
b-2 3 20 150 3 0 1.0 3.8

c-1 3 20 120 1 0 1.0 5.59
c-2 3 20 120 0 1 1.0 7.03

d-1 3 20 150 0 3 2 13.17
d-2 3 20 150 0 1 2

e-1 2 20 100 1 0 1.0 3.995
e-2 2 20 100 0 1 1.0 3.996

f-1 3 30 100 1 0 2.0 4.9
f-2 3 30 100 0 1 2.0 6.0
```

※ オンライン実行環境でデータファイルを扱うには少々面倒な手順が必要なので，このプログラムはパソコンの Spyder，Jupyter Notebook などで実行すること。

★ 繰り返しとリストを使って複数のパラメータに対する解を求める

プログラム 8: ex2021-01.py

```
#!/usr/bin/env python3

for a in [ 1, 2, 3 ]:
    print("a = ", a)
```

プログラム 9: ex2021-02.py

```
#!/usr/bin/env python3

for a in [ 1, 2, 3 ]:
    for b in [ 10, 20, 30]:
        print("a = ", a, "b = ", b)
```

プログラム 10: ex2021-03.py

```
#!/usr/bin/env python3

# x**2 + a*x + b = 0 の解
# 2021/10/30 coded by YAMAMURA,K.

import numpy

print("★ x**2 + a*x + b = 0 の解")
print("")


for a in [ 4, 9, 16 ]:
    for b in [ 1, 2, 4]:
        ans = numpy.roots( [ 1, a, b ])
        # print("a = ", a, "b = ", b, ans)
        print("a = {:2d}, b = {:2d}, 解 {}".format(a, b, ans))
```