

Lesson04 関数

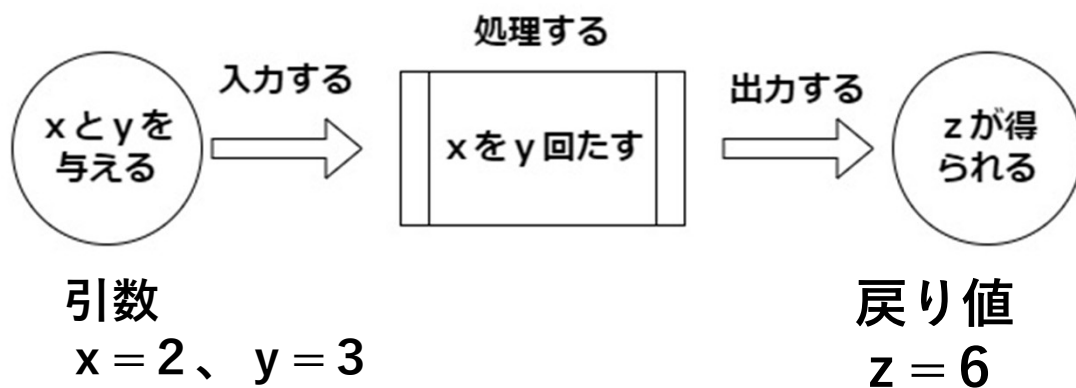
この時間の目的

- ・ 関数を使って、プログラムをわかりやすくする。
- ・ 再帰関数によって数学的問題を解くことが出来るようになる。

1

説明します。 関数とは？

例として、掛け算の関数を考えよう



2

具体的にプログラムを関数を使って書くと

```
# ----- 関数部 -----
# 処理部は関数としてまとめてしまう
# (xをy回繰り返し加えて掛け算を計算している)
def Shori(x,y):          # (x, y)のような入力を「引数」と呼びます。
    kotae = 0
    for l in range(y):
        kotae = kotae + x
    return kotae          # retur命令によって「戻り値」が与えられます。
```

```
# ----- メイン部 の例 -----
# 入力部
x = int(input("xを入力してください:"))
y = int(input("yを入力してください:"))

# 処理部
z = Shori(x,y)

# 出力部
print("掛け算の答えは",z,"です")
```

プログラム
は関数部と
メイン部で
構成します

3

次に実習です。
前の時間に作ったプログラムを関数に改造します。

```
0 1 # 041_func_kaijo_for.py 前出の032_kaijo_for.pyの関数化
0 2
0 3 # --- function ---
0 4 def func_kaijo(n):
0 5     kaijo = 1
0 6
0 7     for i in range(1, n+1):
0 8         kaijo = kaijo * i
0 9     return kaijo
1 0
1 1 # --- main ---
1 2 print(func_kaijo(5))
```

応用問題

前出の階乗の計算プログラム 03_kaijo_while.py も関数化しておきましょう。
プログラム名は 042_func_kaijo_while.py とします。

4

4

```

0 1 # 043_saiki_kaijo.py      n ! を再帰関数を使って求める
0 2
0 3 def factorial(n:int):
0 4     if n == 0:
0 5         return 1
0 6     return n * factorial(n-1)
0 7
0 8 # --- main ---
0 9 n = 5
1 0 print(n,"の階乗は",factorial(n))

```

解説

さらに、再帰関数を使って階乗を計算する関数を紹介します。
 再帰関数とは、自分を呼び出す関数のことです。
 0 3 ~ 0 6 行で定義された関数factorial()の最後の行0 6 行で、自分自身を呼び出していますね。

5

5

```

0 1 # 044_saiki_fibonacci.py   フィボナッチ数列を再帰関数を使って求める
0 2
0 3 def fibonacci(n):
0 4     if n == 1:
0 5         return 0
0 6     elif n == 2:
0 7         return 1
0 8     else:
0 9         return fibonacci(n-1) + fibonacci(n-2)
1 0
1 1 # main (フィボナッチ数列の最初のn個の要素を表示する例)
1 2 n = int(input("フィボナッチ数列の出力個数は？ : "))
1 3 if n <= 0:
1 4     print(" * エラー   :   n は正の整数を入力してください")
1 5 else:
1 6     for i in range(1, n+1):
1 7         print(fibonacci(i), end=" ")
1 8
1 9 print()

```

6

6

フィボナッチ数列は自然が生み出した不思議な数列です。
 大学入試にも出題されているようです。
 高校数学Bの教科書などで学習し、さらには森羅万象の中からフィボナッチ数列を見つけてみてください。

(自然は人智を越えてプログラミング技法を駆使しているらしい)



7

7

```

0 1 # 045_euclid.py   ユークリッドの互除法による最大公約数の計算
0 2
0 3 # ユークリッドの互除法を引き算だけで計算する方法
0 4 def gcd(num1:int , num2:int):
0 5     x = num1
0 6     y = num2
0 7     while x != y:
0 8         if x > y:
0 9             x = x - y
1 0         else:
1 1             y = y - x
1 2     return x
1 3
1 4 # ----- main -----
1 5 num1 = 1071
1 6 num2 = 1029
1 7 print("num1 = ",num1," , num2 = ",num2)
1 8 print("最大公約数 = ",gcd(num1,num2))

```

#応用課題
 #次にユークリッドの互除法を再帰関数で計算します
 #関数gcdを書き換えてみてください

```

def gcd(x , y):
    r = x % y      # x%y の値はx÷yの余りです
    if r == 0 : return y
    else : return gcd(y, r)  # 再帰呼び出し

```

解説

ユークリッドの互除法については高校の数学Aの教科書に基づいて学習し、その実装としてこのプログラムを動かすことが好ましいです。

8

8