令和６年度　ＦＶ・Ⅱ「プログラミングで身近な課題を解決しよう！」

第１回　Python プログラミングの基礎を学ぶ

プログラミングの環境：ノート PC、Jupiter notebook あるいは Google colaboratory

1.変数と値

基本的にx、yなどの変数を宣言してそこに値を代入し計算する形になります。変数名は数行のコードで変数の数も多くない場合は、a,b,c,x,y,z などを使いますが、本格的なコードを書く時は、student_Num、student_Name のように見て内容を理解できる名称にして、他の人にもわかるように書くのがマナーのようです。

２.データ型

プログラミングは、データから何か結果を求めたり、何かを起こさせるものです。まずは、データが基本です。どのようなプログラミング言語でもデータの型（タイプ）やその扱い方を学んだ方が良いでしょう。では、やってみましょう！

実習１：自分の出席番号と氏名、身長を入力する。

■ Num = 43　　#Python では、データの型を宣言する必要がありません。

多くの言語では、最初に　int Num のような型宣言が必要になります。

■ Name = ‘Horiuchi’

■ Height = 173.2

■ print(Num,Name,Height)

#print はコンソールに出力する命令です。何が出力されるでしょう？

■ print(type(Num))

#type はその変数の型を返す関数です。Num の型を出力するという命令

になっていることを理解してください。

■ print(type(Height))

#数値は主に、int(整数)型と float(浮動小数点)型の２つに分かれます。

■ print(type(Name))

#文字は、文字列の str 型に分類されます。

３.コンソール（画面の、>>>　のところ）からの入出力

出力: print( )

入力: input(“ “)　（ “ “の中には、“入力してください>>”）などの言葉を入れる。

input( )によって入力されたものは、すべて文字列(str)型になる。

実習２：数字を２つ読み込みます。その２つの数字の和、差、積を出力するプログラム

■ a = input(“input a”)

= は、 aに代入するの意味です。

■ b = input(“input b”)

（ここを考えて）

■ print(“a+b=” ,x)

■ print(“a-b=” , ・・・・・・・・・・・・・

４.演算操作

（１）数値演算

四則: ＋, ー , * , /,　//(整数の除算切り捨て), %(整数の余り), **(指数)

プログラムの特殊な演算: +=, -=, *=, /=, (a += 1 は、 a = a + 1 という意味になる)

実習３：秒数を入力して、何時間、何分、何秒かを出力する。

■ T = int(input(“input second>>”))

から、H, M, S を求めて

■ print(H, “Hour”)などと出力

（２）文字列演算

基本は　“　”、‘　’ではさむ。

＋　は結合、* は繰り返し。

実習４：ユーザーに名前を入力してもらい、「〇〇さん、こんにちは」

（日本語入力できないようであれば、“ Hello 〇〇！”）と返す。

５.配列型のデータ

Python では配列（個々のデータの集合体）もデータになる。配列のデータにもいくつか種類があるが、基本的でよく使われるものを紹介する。

（１）リスト　list

[1,2,3,4]のように[]で囲まれ、“　”で囲まれる。中身の個々のデータを要素と呼び、要素は、int 型、float 型、string 型など何でも良い。

実習５：Greeting リストを作成、要素の取り出し、書き換え、削除、追加などを行う。

■ Greeting = [‘Good morning’, ‘Good afternoon’, ‘ Good bye’]

■ print(Greeting) として実行し、中身を確認する

■ Greeting[1]は？

■ 書き換え:Greeting[2] = ‘Sayounara’

■ Greeting.append(‘Good night’)

■ Greeting.remove(‘Good night’)

（２）Numpy 配列

科学計算に強いライブラリ（小プログラムの集合）である Numpy には、narray という配列がある。理数学的には、ベクトルや行列を意味している。

実習６：Numpy 配列を作って演算をしてみる。

■ Import numpy as np

■ a = np.array([1,2,3])

■ b = np.array([4,5,6])

■ a + b

■ a * 2

■ a * b

よく使われるのは、

0（ゼロ）～ n-1 までの整数を並べる np.arange(n)

0（ゼロ）を並べる　np.zeros(n)

１を並べる　np.ones(n)

乱数を並べる　np.random(n)

■ a = np.arange(10)

■ b = np.zeros(10)

■ c = np.ones(10)

■ d = np.random.random(10)

■ e = np.random.randint(10)

令和５年度　FV「RaspberryPi を用いた IoT プログラミング実習」

第２回　じゃんけんプログラムをつくる（ＡＩ?）

プログラミングの環境：ノート PC、Jupiter notebook あるいは Google colaboratory

準備１：出す手の表示コード

■ # じゃんけんプログラム(名前)　‘#’をコメントアウトといいます。

■ Hands = [“GU” , “CHOKI” , “PA”]　# グー、チョキ、パーのリスト

準備２：人間（自分）が出す手を決める

■ Choice_Human = int(input(“0 から2の整数を選んでください>>”))

■ print(Hands[Choice_Human])　# Hands_Human[0]は、”GU”みたいな･･･

準備３：相手の出す手をコンピュータに決めてもらう

■ import random　　# 内部ライブラリ random のインポート

■ Choice_PC = random.randint(0,2)　# 0～2をランダムに選ぶ。

■ print(Hands[Choice_PC])

準備４：Human と　PC　のじゃんけんの勝敗を判定する

■ # じゃんけんプログラム(コンピュータ)

■ if Hands[Choice_Human] == “GU” and Hands[Choice_PC] == “CHOKI”:

　〇〇〇〇print(“あなたの勝ちです“)

■ elif Hands[Choice_Human] == “CHOKI” and Hands[Choice_PC] == “GU”:

　〇〇〇〇print(“あなたの負けです“)

☆ すべての場合の判定ができるかな？

```
Hands = ['GU','CHOKI','PA']
Choice_Human = int(input('0から2の整数を選んでください＞＞'))
print(Hands[Choice_Human])
import random
Choice_PC = random.randint(0,2)
print(Hands[Choice_PC])

if Hands[Choice_Human] == Hands[Choice_PC]:
  print('引き分けです')

elif Hands[Choice_Human] == Hands[Choice_PC-1]:
  print('あなたの勝ちです')

elif Hands[Choice_PC] == Hands[Choice_Human-1]:
  print('あなたの負けです')
```

第３回　プログラミングの基本構造を絵を書きながら学ぶ

～　Turtle を使った実習～

１．　ColabTurtle の起動

■ ! pip3 install ColabTurtle

■ from ColabTurtle.Turtle import *

と入力して　Shift ＋　Enter

２．　亀の初期化

■ initializeTurtle(initial_speed =7)

■ pencolor(‘red’)

‘yellow’ など好きな色にしてみてください。

３．　主なコマンド

■ forward(50)　　# 前に50進ませる

■ right(100)　　#亀の角度を100度右に回転させる

■ left(60)

４．　「繰り返し文」を学ぶ

■ for i in range(3):

　　○○○○　forward(50)

　　○○○○　right(100)

問　正多角形や星を Turtle を用いて書いてみよう！！

令和５年度　FV「RaspberryPi を用いた IoT プログラミング実習」

第４回　外部モジュール matplotlib を用いて、グラフを表示させてみよう！

お題１

```
■ import matplotlib.pyplot as plt      #  matplotlib  を  pltで表す
■ x  =  [ 1 , 3 , 5 , 7, 9 ]          #  x 軸の値を入力
■ y  =  [ 2 , 4 , 6 , 8 , 10 ]        #  y 軸の値を入力
■ plt.plot( x , y )                   #  グラフ表示させるコマンド
■ plt.show(  )
```

お題2

```
#  規則性のあるリスト(配列)を「繰り返し文」にて作製する方法
■ x  = [ ]              #  入れ子をつくる。
■ a  = 0                #  変数 a の初期設定
■ for i in range(10):               #  繰り返し文にて、x 軸を生成する
      a  =  a  +  1
      x.append(a)
■ print(x)
```

お題3

```
#  ライブラリ  numpy  を用いると、もっと簡単に、リスト(配列)をつくれる！
■ import numpy as np
■ x = np.arange(0 , 10 , 0.2)
■ print(x)
```

お題4

＃　numpy 配列は、とても便利です。

よく使われるのは、

　0（ゼロ）～ n-1 までの整数を並べる np.arange(n)

　0（ゼロ）を並べる　np.zeros(n)

　１を並べる　np.ones(n)

　乱数を並べる　np.random(n)

■ a = np.arange(10)

■ b = np.zeros(10)

■ c = np.ones(10)

■ d = np.random.random(10)

■ e = np.random.randint(10)

■ print(?)　　＃　?に、a～e を入れて、出力してみよう！

お題 5

＃　ライブラリ　numpy　で軸の値のリストをつくり、matplotlib でグラフを書く。

■ import matplotlib.pyplot as plt

■ import numpy as np

■ x = np.arange(0 , 10 , 0.5)

■ y = np.sin(x)

■ plt.plot(x,y)

■ plt.show()

第５回　円周率πをプログラミングで求める！～モンテカルロ法を用いて～

プログラミングの環境：ノート PC、Jupiter notebook あるいは Google colaboratory

■ import random　　＃　random モジュールをインポートする

■ import matplotlib.pyplot as plt　　＃　matplotlib モジュールをインポートする

■ totalcount = 100　　＃　試行回数を設定する。変更可能。

■ incount = 0　　＃　変数を設定。

＃　x 軸、y 軸の第 1 象限にランダム関数の(x,y)の座標をプロットする。

■ for i in range(totalcount):　　＃　0 から順に整数値を i に100回代入しする。

```
    x = random.random()　　＃　0から1までの小数(float 型)をランダム関数で生成
    y = random.random()　　＃　0から1までの小数(float 型)をランダム関数で生成
    if x**2 + y**2 <= 1.0:
        incount = incount + 1　＃　条件を満たす時、incount の値を随時、更新する
        plt.scatter(x , y , c = ‘ red ‘)
    else:
        plt.scatter(x , y , c = ‘ blue’ )
```

■ print(‘pi :{ }’ .format(　　？　　) ＃　？には、円周率πを求める演算式を書く！

```
plt.show()
```

第６回　「素数」か否かを判定するプログラミングのコードを書く！

プログラミングの環境：ノート PC、Jupiter notebook あるいは Google colaboratory

Q．素数とは何ですか？

お題1　自分で入力した整数が「素数」かどうかを判定するプログラムを書く！

■ a = int (input(‘お好みの整数を入力してください>>’))　# str型の a を int 型に変換

■ b = ‘ { }は、素数である’．format(a) ‘　　# int 型のaを文字列に変換し、{}に代入

■ for i in range(？, ？, 1):　　# range()のリスト(配列)の意味を確認する。

　　if a　　:　　# a が素数ではない条件は何？

　　b = ‘ { }は、素数ではない’．format(a) ‘　# a が素数ではない条件は何？

■ print(b)

お題２　1～1000までの整数が、素数かどうかを記述するプログラムを書く！

```
for a in range(1000):
  b = '{}は、素数である'.format(a)
  for i in range(2,a,1):
    if a % i == 0:
      b = '{}は、素数ではない'.format(a)
  print(b)
```