「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ぼう

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)