Pythonxn使用法

はじめに

Pythonにはver.2系列とver.3系列がある

Ver2系列 現在最も多く利用されている。Unocode関係が使いづらい。

print u"日本語" #２バイト系の文字コードの指定

Ver3系列 内部的にunicordに完全対応。

print文がカッコが必要になる。

print ("日本語") #２バイト系の文字コードの指定

情報はPython2の方が多いのでここでは２系を使用する

Pythonを科学技術で利用して分析作業をやりやすくするパッケージがpythonxyである。

pythonxyをインストールすると、

ipython pylab scipy matplotlib sympy numpy などのPythonパッケージ、

アプリケーションとしては、console,scite,spyderなどがインストールされる。

console ターミナルを複数実行できる。

scite コード補完などのできるエディター

spyder pythonの統合開発環境(IDE)

配布したファイルを解凍してできるフォルダをエクスプローラーで開き、右クリックメニューから、Open enhanced console hereを実行する。Consoleというアプリケーションが起動する。左端のツールボタンからIPyson(Qt)を選択すると新しいタブができる。このタブでpythonを実行できる。

1+1と入力しENTERで答えの２が表示される。

sin(30\*pi/180) と入力しENTERで0.49999999999999994が表示される。

range(0,5)と入力すると[0,1,2,3,4]が表示される。

x=arange(0,2\*pi,0.1) xに0-2πの配列データが0.1刻みで入る。

y=sin(x) yにsin(x)を計算したデータが入る。

plot(x,y) グラフが表示される。

演習　y=x2-1のグラフを-5<x<5の範囲で作成しなさい。

xのn乗はx\*\*nと書く。（Python,Ruby,Perlでx\*\*n 　C、pascal、Basicはx^n、R両方）

1/2と入力すると答えは0.5ではなく０になる。これは２系のPythonの特徴で

1/2の1と2は整数型なので答えも整数型になるので切り捨てになった結果こうなる。

1.0/2と入力すると答えは0.5となる。この場合は1.0がfloat型で答えもfloat型になるので。

演習　1/2　+　1/2の答えは？正しく計算するにはどうするか。

演習　1==1を入力すると答えはTrueになる。3.3-3.2==3.2-3.1の答えを確かめる。

scite p0 と入力しTabを押すと入力候補が表示される。

scite p001まで入力してタブを押すとscite p001.pyが補完される。Enterで実行する。

SciTEというエディターが起動しpoo1.pyが読み込まれる。

-------------------------------------------------

# -\*- coding: utf-8 -\*-

"""

Ploject Euler Problem 1

10未満の自然数のうち, 3 もしくは 5 の倍数になっているものは

3, 5, 6, 9 の4つがあり, これらの合計は 23 になる.

同じようにして, 1000 未満の 3 か 5 の倍数になっている数字の合計を求めよ.

"""

-------------------------------------------------

これはPloject Euler の問題１の日本語訳で、Pliject Eulerは数学の問題を公開しているサイトで現在５００ほどの問題があり、プログラミングの教材に使用されている。

この問題をPythonで解答して見よう。

最初に３か５の倍数になっているかどうか確かめる関数を作成する

def f35(x):

if x%3==0 or x%5==0:

return True

else:

return False

print f35(11)

print f35(12)

予約語や数値などが自動的に色分けされる。また改行すると自動的にインデントされる。

メニューからTooLsでGoを選択すると下のコンソールウインドウで実行される。実行に成功すると　False　True　>Exit code 0　が出力される。

def f35(x): 関数宣言

if x%3==0 or x%5==0: | x%3 はxを３で割った余り

return True | Trueは１

else: | Falseは０

return False |

ans=0 答え用の変数の初期化

for i in range(1,1000): Forループ

if f35(i): |Pythonのループは

ans+=i |　for 変数　in リスト　と記述する。

print i |　range(a,b)で　[a,a+1,a+2,...,b-2,b-1]のリスト

print "ans=", ans 　　print a,bでaとbが標準出力に出力される。

p002.py

----------------------------------------------------------------------

# -\*- coding: utf-8 -\*-

"""

Ploject Euler Problem 2

フィボナッチ数列の項は前の2つの項の和である.

最初の2項を 1, 2 とすれば, 最初の10項は以下の通りである.

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

数列の項の値が400万以下の, 偶数値の項の総和を求めよ.

"""

----------------------------------------------------------------------

フィボナッチ数を求める関数fibの例。

def fib(x):

if x==1:return 1

if x==2:return 2

return fib(x-1)+fib(x-2)

print fib(8)

これは関数の中で自分自身を使用するもので再帰関数という。現代的な言語で使用できる。

今回は大きなフィボナッチ数を求めるのでこの関数では遅い。

p002.py

----------------------------------------------------------------------

省略

n=400\*10000

a=1

b=2

ans=0

while True:

c=a+b

print c

if c%2==0:

ans+=c

a,b=b,c

if c>n:

break

print "ans=",ans

----------------------------------------------------------------------

p003.py

----------------------------------------------------------------------

# -\*- coding: utf-8 -\*-

"""

Ploject Euler Problem 3

13195 の素因数は 5, 7, 13, 29 である.

600851475143 の素因数のうち最大のものを求めよ.

"""

from sympy import factorint

n=600851475143

----------------------------------------------------------------------

コード部分に入力されている。fromはモジュールの読み込みの指定で、標準に読み込まれていない関数などを利用できるようにする。

よく使用されるモジュールにはnumpy、matplotlib,、sympy、pylab、scipyなどがある。

from numpy import sin でsinが利用できる。

from numpy \* でsin cos logなどの関数が使用できるようになる。

import numpy でnumpyの関数が利用できるが関数名は numpy.sin　になる。

import numpy as np のようにすると関数名はnp.sinになる。

\*違うモジュールで同じ関数名を使用している場合があるので注意（numpyとsympyのsinなど）

from pylab inport \* では科学技術用にnumpyやmatplotlibがインポートされsinやplotなどの関数が利用できる。

from sympy import \*では分数の計算、積分、微分、因数分解、式の簡略化などができる代数演算環境になる。

p003.py

----------------------------------------------------------------------

# -\*- coding: utf-8 -\*-

"""

Ploject Euler Problem 3

13195 の素因数は 5, 7, 13, 29 である.

600851475143 の素因数のうち最大のものを求めよ.

"""

from sympy import primefactors

n=600851475143

a=primefactors(n)

print a

print max(a)

----------------------------------------------------------------------