JAPLA 研究会資料　2021/2/15

Python プログラミングへの J ユーザの体験記

西川　利男

最近は Python が真っ盛りだと、志村正人氏がコメントされている。なるほど、書店のプログラミング言語のコーナーを見渡すと、C や Java などと並んで Python の本が何冊も並んでいる。

言語仕様などは C, Java などと比べて、整数、浮動小数点数の値の型宣言や行末のセミコロンが不要など、BASIC 並みでありこの点は好感がもてる。

しかし、プログラミングのシステム環境－使い勝手は実際に動かして見ないとわからない。Windows 7 のマシーンの上で何とかインストールして、いくつかのプログラムを動かして見た。J ユーザの一人としての私の個人的な体験を報告する。

なお、主に参考としたのは Python のオンラインマニュアルと下記の書である。

[1] Peter Farrel 著、鈴木幸敏訳「Python ではじめる数学の冒険－プログラミングで理解する代数、幾何学、三角関数」オライリー・ジャパン、オーム社(2020).

１．Python を動かす２つの方法

APL や J では、システムを立ち上げると、入力してすぐ実行する実行環境が現れる。（J では ***. ijx）　それで良いとなれば、ファイルとして保存する(***.ijs)。

ところが、Python ではこれとはすこし異なり、最初は戸惑った。

Python はあくまで言語であり、それを動かす環境はそれとは別にある。

従って、直接法とエディター環境から Python を呼び出し実行する方法とがある。

・直接法

Python のキーワードを入力して、その結果を得る。

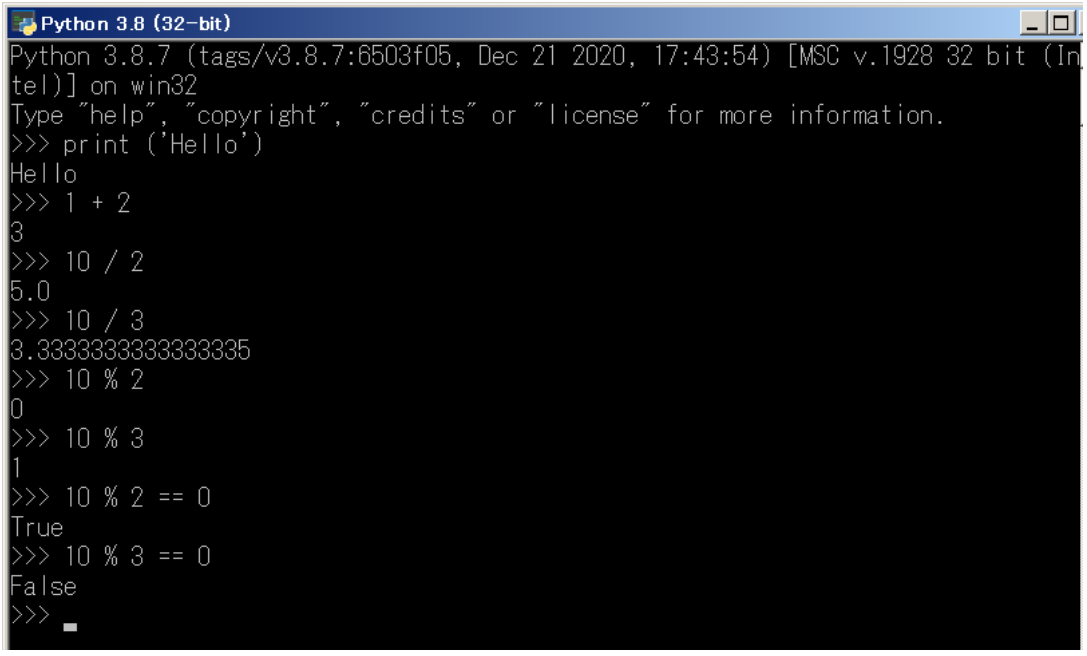
・Visulal Studio Code を用いる方法

たとえば一種のエディタである Visual Studio Code なるシステムを起動して、その中で言語 Python でプログラムなど実行コードの集合、つまりスクリプトを作る。あらためてその環境のもとで、Python を起動して、実行する。

以下、それぞれの方法を見ていこう。

２．Python の直接実行法

Python を直接起動すると、下のような黒い画面が現れる。
そして ＞＞＞ のように入力プロンプトがしめされるので、それに続けて入力すると、ただちに実行されて、結果が表示される。



いくつかの簡単な例をやってみよう。print により文字列の表示がなされるが、かっこでくくることが必要である。

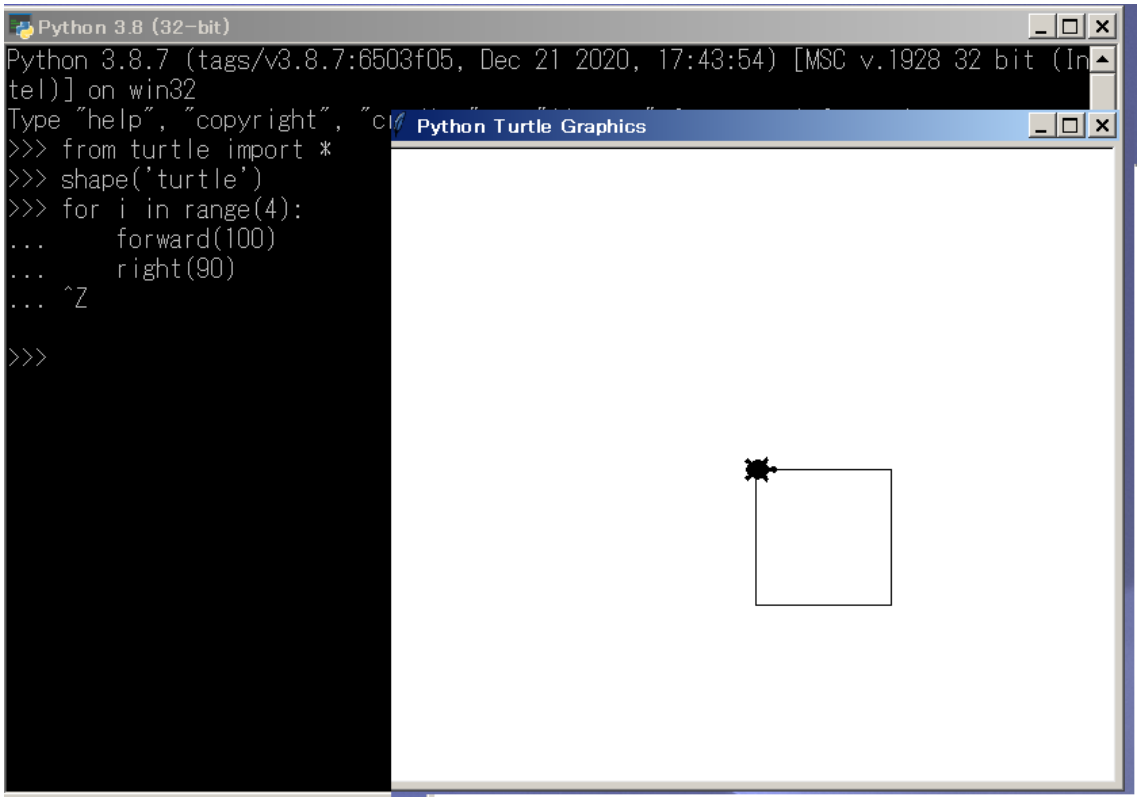
数値計算は気楽に行える。割り算は記号 / により自動的に浮動小数点数で結果が返えされる。割り算の余りは記号　% で行われる。等しいかどうかの真偽は True, False で返される。演算処理は左から右への順であり、J ユーザにとっては、かえってとまどるかもしれない。ちなみに関数型言語 J では演算は右から左へと行われる。

次に繰り返しループを使った複数行の例をあげてみる。CTRL-Z で実行する。

```
>>> for i in range(10):
...     if i % 2 == 0:
...         print(i)
... ^Z

0
2
4
6
8
>>>
```

ファレルの本から、Python でタートルグラフィックスが簡単に行える記載があったのでやってみる。import により導入して、亀の絵で正方形を描いてみた。



直接の実行法では、結果がすぐ返され、電卓のように手軽である。しかし、長い複雑な処理を行うときにはその度に入力しなければならないので、やはり大変である。

３．Visual Studio Code を用いて Python を実行する方法

Visual Studio Code はメモ帳などと同じく一種のエディタである。その中に Python のプログラムテキストを書いてファイルとして保存する。

その後、この中から Python の言語プロセッサを呼んで実行する。ちなみに Python に限らず、C や Java などいろいろな言語を呼んで実行することができる。

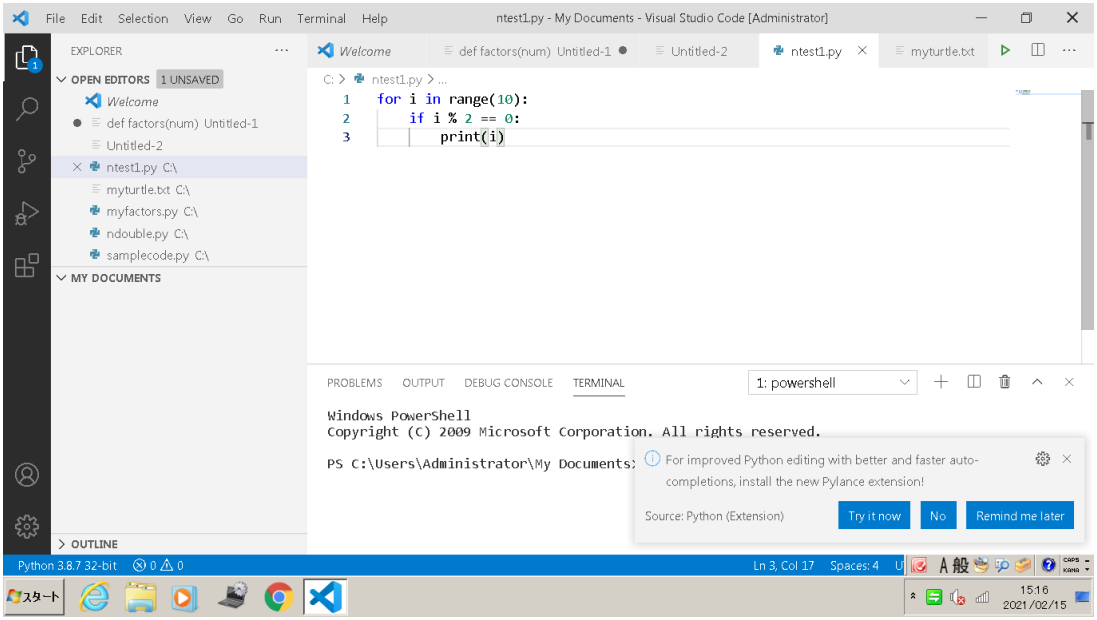
Visual Studio Code による実例はつぎのようになる。

画面が開いたら、プログラムテキストを入力して、名前をつけて保存する。名前は

***.py

とする。

ここでは、ntest1.py とした。



プログラムの実行は、

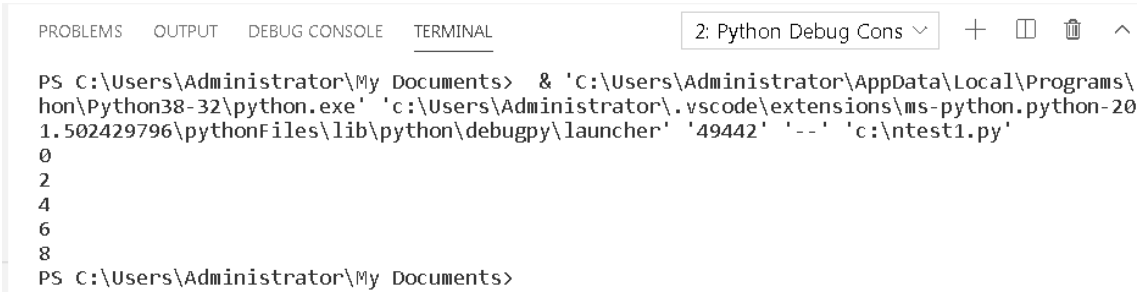
[Run] – [Run Without Debugging]　をクリックするか

または

CTRL-F5

として行う。

すると、画面の下半分に、以下のように結果が表示される。



４．おわりに

Python はシステムではなく、言語プロセッサである。以前 Processing により、Java を言語とするグラフィックスを報告したが[2]、これを Python コードで行うことも良く行われる 。Python は通常の言語システムとは違うことが分かった。

[2] 西川利男「Processing プログラムをトライしてみようーJ ユーザから見た使い易さ、難さの体験ー」JAPLA 研究会資料 2020/10/17

はじめに Python はセマンティックス、シンタックスの両方について、手軽で良いと述べたが、いくつかの点が気にかかった。

行末の CR が言語の処理の上で意味をもっている。つまり見やすさだけで改行することはできない。さらに強制的なインデンション（桁下げ）はうっとうしい。この点では、C, Java のように改行は単なる見易さだけであり、処理構文の区切りはセミコロンでという方式はそれなりの意味を持つ。

わたくしの個人的感想としては、Python では自らプログラミングを行うというより import により手軽に資源を利用するという使い方を行うものだろうと感じた。