

Python の導入-JupyterNotebook とライブラリー

SHIMURA Masato

2018 年 8 月 7 日

目次

1	Jupyter Notebook のインストールと立ち上げ	2
2	Python -JupyterNotebook 入門	3
3	Numpy を使う	4
4	Pandas を使う	5
5	matplotlib	6
6	Markdown	7
7	JupyterLab を使ってみる	7

Python はオランダ人ガイド・ヴァンブロッサムが 1991 年に版を公開した。ABC 言語をベースに暇に任せてちょこちょこ作成して、大好きなモンティ・パイソンから名付けたようである。

2005 にヴァンブロッサム自身が Google に雇用され、Google の公式言語の一つになっている。

世界の腕自慢 90 万人が参加する数論とプログラムのページ・ProjectEuler

<http://projecteuler.net>

では C 連合 (C/C++/C#/Java) と双耳峰を構成している汎用言語であり、最近の統計でも JAVA を追い越したようだ。

比較的新しい言語で手軽に使えるインタプリッタであり、概ね新しい機能は全部搭載しており、AI から機械制御まで守備範囲は広い。個人が開発した言語でオブジェクトに凝り固まっているが、すっきりしたソースコードを手軽に提供できる

IBM やマイクロソフトは言語を提供しても、その後は有償として困り込んだ。新しい覇者 Google はユーザーとコンテンツを重視するので豊富な資源を解放している

- 伝統的なプログラム作成
- 豊富なライブラリーを利用した計算ツール
- AI などの先端ライブラリーの利用。
- 電子工作のプログラムツール

1 Jupyter Notebook のインストールと立ち上げ

Anaconda は Jupyter Notebook や Jupyter Lab など多くの環境を提供しており、VSCode も Anaconda 上で使える

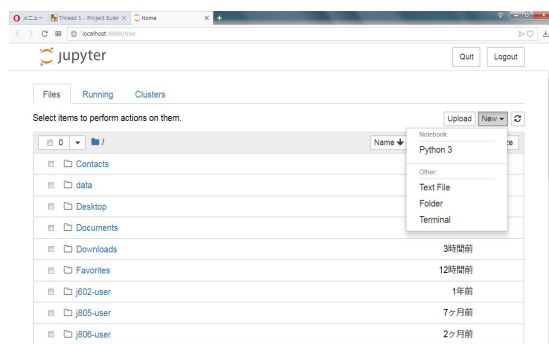
python 用の環境であった Jupyter Notebook は今では多くの言語に対応し、WorkSpace を提供する。

APL はワークスペースを用いる言語なので馴染みがあるが、暫く使わないとゴミ箱のようになってしまう。

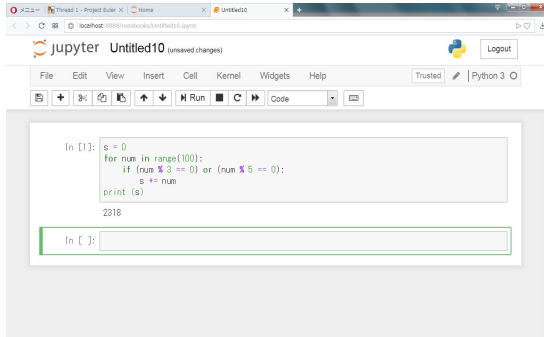
J はワークスペースを廃止したが、時代は巡ってまた WS が先端になっているようだ。

(J はエディタが付いているのでそれほど感激しないが)

1. anaconda をインストールする。(anaconda で検索すると幾つか出てくる)
<https://www.anaconda.com/download/>
2. OS を選ぶ。Python は Var.3 を選ぶ
3. DL してインストールする
4. WINDOWS ボタンで [anaconda navigator] を立ち上げ [JupyterNotebook] を選ぶ (Launch)
5. JupyterNotebook が立ち上がったなら右上の [New] で [Python3] を選ぶ



- ソースコードは [Code] 窓に書き込むか、別に用意する Run ボタンで実行する



- File Open で読み込んだソースコードの処理方法？
- ワークスペース環境でもソースコード単独でもセーブ出来る
- Code モードとメモを記述できる [Markdown] モードもある

Mathjax をサポートしており [Tex] で数式が記述できる

- WS の出力は他の PC デモ利用できる。

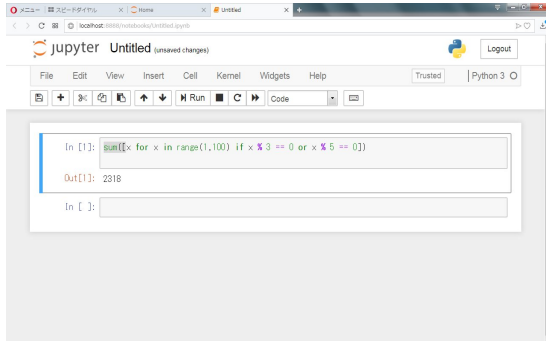
2 Python -JupyterNotebook 入門

1. Code を入れてみる。(Copy & Paste)

```
answer = 0
for i in range(1,101):
    answer += i
answer
```

[Out] 5050

- オリジンは 0 のようで 1 番目で 0 をとる。0 から初めても答えは同じなので 0 は無駄打ちか。
- : end ではない
- 最終行の answer は左詰め的位置が必要



3 Numpy を使う

数値演算モジュール *numpy* を使ってみる。

NumPy とは配列環の通っていない *Python* の数値計算アタッチメント。*APL* や *J* ユーザーにとってはダックスフントをつれて散歩をするようなもの。

ほとんどは *J* の関数や簡単なイディオムで対処できる。紹介していると本が一冊出来てしまうので [*Python*] [*NumPy*] で検索して調べてほしい。

1. 最初にインポート
2. オブジェクト云々は数学には無用なので触れないが何気に忍び寄られている

```
import numpy as np
```

- *Run* を押してエラーが出なければ ロードされている。
- *np* は ティアラのようなもので何でも良い。(*np* が相場)

3. 内積計算

```
ar1 = np.array([[1,2],[3,4]])  
ar2 = np.array([[5,6],[7,8]])  
np.dot(ar1,ar2)
```

```
[out] array([[19,22],  
            [43,50]])
```

4 Pandas を使う

1. JでCSVファイルを作る

```
a=. ? 10 3 $ 100
require 'csv'
a writecsv 'c:\temp\test.csv'
99
```

2. pandas をロード

```
import pandas as pd
```

3. csv ファイルの読み込み

```
data = read_csv('c:/temp/test.csv')
print(data)
```

```
      31  81   7
0  42  23  45
1  95  20  23
2  87  25  62
3  63  76  45
4  83  60  97
5  75  88  89
6  84  84  90
7  93  78  31
8  24  91  99
```

4. DataFrame を作らないといけないようだ。

5. csv ファイルに直接書いた方が早い

```
data = read_csv('c:/temp/test2.csv')
print(data)
```

a,b,c	a	b	c	
31,81,7	0	31	81	7
42,23,45	1	42	23	45
95,20,23	2	95	20	23
87,25,62	3	87	25	62
63,76,45	4	63	76	45
83,60,97	5	83	60	97
75,88,89	6	75	88	89
84,84,90	7	84	84	90
93,78,31	8	93	78	31
24,91,99	9	24	91	99

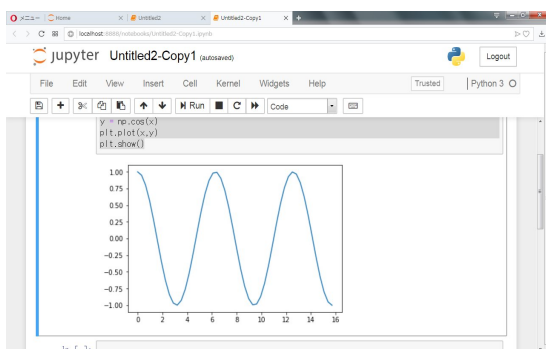
6. *DataFrame* をしっかり設計すると、*EXCEL,CSV,DB* のデータを組み合わせて使用することも出来る

5 matplotlib

グラフ表示ライブラリを使ってみる

```
import matplotlib.pyplot as plt
# inline
%matplotlib inline
import math
import numpy as np
x = np.linspace(0 , 5 * math.pi)
y = np.cos(x)
plt.plot(x,y)
plt.show()"
```

#がコメント



6 Markdown

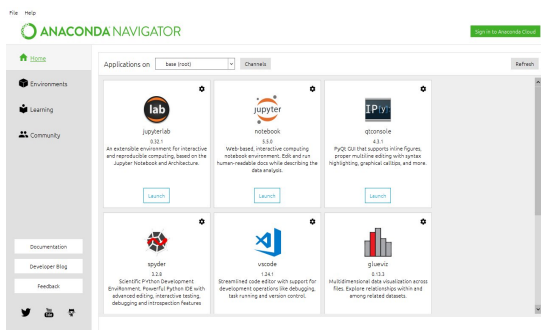
`python` の上真ん中にある「Code」と表示されているダイアログボックスで [Markdown] を選ぶとメモ帳モードに切り替わる。共用することもあるのでわかりやすく書いておこう。

- # あいうえお --> section
- ## かきくけこ --> subsection
- * なになに --> item1
- * いろいろ --> item2
- \$ 3 \times \pi \$ --> TeX で数式を記述できる $3 \times \pi$

7 JupyterLab を試してみる

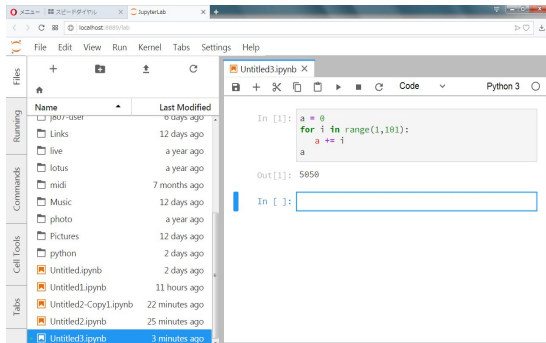
JupyterLab は Jupyter Notebook の進化形として開発が進められており、2018 秋に Ver1 が出る予定

NoteBook と同じくアナコンダから立ち上げる



`python3` をクリックすると Notebook と似た画面が出てくる。実行ボタンは右三角に簡略化

Jupyter Notebook と比べるとプログラマーには他人の家とシェルコマンドが使える我が家の差か。



References

python に関する例題の結果の多くは次を動かした結果である
中島省吾「人工知能プログラミング入門」日経ソフトウェア 2018/05号：付録