

スマホを関数電卓で活用する方法

- モバイル用 Python アプリについて -

増山 幸一
明治学院大学経済学部

2015年2月: preliminary version

1 スマホまたはタブレット用 Python アプリのインストール

スマホ用の関数電卓は多数ありますが、Python はプログラムを書いて保存できるので、それらよりもはるかに高機能な能力を発揮します。Python をスマホにインストールして、効率的に活用する方法を説明します。

スマホあるいはタブレットに Python をインストールするためには、Python アプリをダウンロードする必要があります。Android 系モバイルでは、「QPython 3」をインストールすると便利です。無料でダウンロードできますので、Google play ストアから「QPython 3」をダウンロードして下さい。

iOS では、「iOS 用の Python3.3」がシンプルで便利です。(iOS のソースコードが未公開なので、iOS 用のアプリは Android 系に比較して有料のケースが多いです。)「iOS 用の Python3.3」は有料で、100 円です。100 円の価値は十分ありますので、App Store から購入して、ダウンロードして下さい。(「iOS 用の Python3.4」は 200 円ですが、3.3 の方で十分です。)

これらの 2 種類の Python アプリには、テキスト・エディターが内蔵されていて、日本語の処理も可能になっています。スマホのエディターで Python のスクリプトファイルを作成して保存することが容易にできます。

スマホの Python アプリは Python 本体だけがインストールされていて、複雑な数値計算用の関数や高度な作図機能を持つモジュールは組み込まれていません。行列を用いた高度な数値計算やグラフィックスを必要としない限り、関数電卓として活用する目的には十分です。

2 Python console の使用方法

最初に、会話型シェル上のプロンプトの後に命令文を入力して計算する直接的な方法について説明します。Python アプリを起動したら、console と書いてある箇所をクリックして下さい (iOS では、画面下の左端にあるパイソンマークをクリックして下さい)。すると、下のような (以下の表示は OS X 用の Python 3.4 なので同じではないですが、類似の) 画面がでます。

```
*****
```

```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 5 2014, 20:42:22)
[ GCC4.2.1 (Apple Inc. build 5666) (dot 3) ] on darwin
Type "copyright", "credits" or "license()" for more information.
```

```
>>>
```

```
*****
```

これで Python を使用可能な状態、つまり、コマンドが入力できるようになります。このコンソール上で、プロンプト (>>>) の後に Python のコマンド (命令文) を入力して操作します。この画面を Python shell、その機能をインタプリタと言います。すべてのコマンドの入力は英文入りに切り替えてください。最初に、

```
>>>x=12.5
```

と入力して下さい。変数 x に数値の 12.5 が格納されます。ここで、x の内容を確認したいので、

```
>>>x
```

と入力すると、x の内容が表示されます。確かめて下さい。インタプリタは命令文を順次一つずつ実行して行きます。間違いがあると、そこでエラーを出して処理を停止して、次のコマンド入力を待ちます。

コンピュータ内では数値は 2 進法で処理されるため、厳密な小数点数値の処理はできません。おそらく、スマホの CPU 能力からすると、小数点以下 15 桁から 18 桁目がゼロではない数値になる場合が多いと思います。文字列はコンピュータ内部では unicode の体系コード表に従って (utf-8 形式で) 符号化され、保存・処理されます。

次に

```
>>>a='Hello World'
```

と入力して下さい。変数 a という容器に文字列の Hello World が格納されます。文字列を格納するときは、上のように文字列を引用符で括ります。引用符として "Hello World" も使用できます。

```
>>>a
```

と入力して下さい。'Hello World' と表示され、プロンプトが返されます。日本語の文字列を取上げましょう。日本語を入力するときは、日本語入力モードにキーボードを変えて下さい。ただし、日本語以外の変数名などのコマンドはあくまで半角英文モードで入力して下さい。

```
>>>b='朝です！'
```

```
>>>b
```

と入力してみて下さい。どうですか、'朝です！' が表示されましたか？ このように定義される x、a、b を変数と言い、x や a、b の内容物をオブジェクトと言います。Python Console を終了するときは、

```
>>>quit()
```

と入力して下さい。この命令文を入力しないと、background で Python が入力を待っている状態になります。

3 インタプリターを用いた数値計算

変数を処理するに当たり、使用可能な四則演算子は以下の通りです。

足し算	+	$2.0+4.5=6.5$
引き算	-	$2.0-4.5=-2.5$
掛け算	*	$2.0*4.5=9.0$
割り算	/	$2.0/4.5=0.4444\dots$
累乗	**	$2.0**4.5=22.6274\dots$
余り	%	$4.5\% 2.0=0.5$

次に、この演算子を使った計算をしましょう。

金融論の最初に登場する投資プロジェクトの現在価値の計算式を取上げる。1年後にキャッシュフロー C として 100 万円を生み出す投資プロジェクトの現在価値は

$$PV = \frac{C}{1+r}$$

で計算される。ここで、 r は割引率である。割引率が年率 5% であるとき、つまり、 $r = 0.05$ のとき、現在価値は

```
>>>r=0.05
```

```
>>>pv=100/(1+r) #単位は万円 (>>>100/(1+0.05) と書いても良い)
```

```
>>>pv
```

と入力すれば、結果が表示されます。結果は 95.23... となります。2 年後に $C = 120$ 万円を生み出す投資プロジェクトの現在価値は

$$PV = \frac{C}{(1+r)^2}$$

と計算されるので、

```
>>>pv=120/(1+r)**2
```

```
>>>pv
```

と入力します。結果が、108.84... と表示されます。今後 3 年間に渡り、年 $C = 120$ 万円のキャッシュフローを生み出す投資プロジェクトの現在価値は

$$PV = \frac{C}{1+r} + \frac{C}{(1+r)^2} + \frac{C}{(1+r)^3}$$

と計算される。なので、

```
>>>c=120
```

```
>>>pv=c/(1+r)+c/(1+r)**2+c/(1+r)**3
```

```
>>>pv
```

と入力する。結果は 326.78... です。コンピュータ内部での四則演算子の計算では、() の計算が最優先され、次に、べき乗**が優先されます。次に、掛け算と割り算が行われ、最後に、足し算と引き算が実行されます。

数値計算を効率的に行うには、Python の標準ライブラリの math モジュールを活用するのが便利です。

math モジュールには、三角関数、指数・対数関数などが組み込み関数として準備されています。math モジュールを使用するためには、使用する前に、math モジュールを import する必要があります。つまり、

```
>>>from math import *
```

と記述しておく必要があります。例えば、2.56 の（自然）対数値を求めるときは、

```
>>>from math import *
```

```
>>>log(2.56)
```

と入力すれば良い。0.94000...と答えが表示されれば成功です。

ファイナンスでは金利計算が連続複利で計算されるケースが多い。1 億円の投資をして、金利が年率 5 % であるとき、1 年後の累積額は連続複利で計算すると、 $e^{0.05}$ となる。これを計算するために、上の入力に続いて

```
>>>exp(0.05)      (>>>e**0.05   でも良い)
```

と入力する。結果は 1.0512...と表示される筈です。math モジュールの import は最初に一度書いておけば良い。

import の入力形式を

```
>>>import math as ma
```

と書いても良いですが、このような形式では、math モジュールの組み込み関数 function の名前の前に必ず ma.function と書く必要があります。この場合

```
>>>ma.log(2.56)
```

のように書かなければいけない。この煩雑さを避けるためには、上のように書いた方が便利です。

(Python 3.4 以降では、標準モジュールに数理統計用の statistics モジュールが追加されていますが、スマホなどの Python App には付属していません。)

4 スクリプトファイルを用いた計算

次に、単なる電卓ではなく、プログラムを作成して、それをスクリプトファイルとして保存して活用するという高度な利用方法を説明します。1 年後 1 億円、2 年後に 1.2 億円、3 年後に 1.3 億円、4 年後に 1.1 億円、5 年後に 0.9 億円の収益が見込まれる投資プロジェクトの現在価値の計算を取上げます。投資資金が 4 億円必要で、割引率が年率 5 % であるとします。

初年度には 4 億円の支出が必要なので、現時点でのキャッシュフローはマイナスなので、-4 億円となります。この投資プロジェクトのキャッシュフローの流れを、 $C = [C_0, C_1, C_2, C_3, C_4, C_5]$ と表現し、この投資プロジェクトの現在価値を計算するプログラムを作成する。具体的な数値としては、

$$C = [-4.0, 1.0, 1.2, 1.3, 1.1, 0, 9]$$

となっています。このとき、現在価値の計算式は以下の通りです。

$$PV = C_0 + \frac{C_1}{1+r} + \frac{C_2}{(1+r)^2} + \frac{C_3}{(1+r)^3} + \frac{C_4}{(1+r)^4} + \frac{C_5}{(1+r)^5}.$$

この計算をするために、スクリプトファイルを作成します。

スマホで python アプリを起動して、エディターを開いて下さい。iOS では、画面下の左から 2 番目のアイコンマークをクリックして下さい。エディターが開きます。

以下のスクリプトで、#から始まる文はコメントであることを示します。複数のデータを一つの変数に格

納するためには、各データをコンマで区切って、鍵括弧で括弧します。下のスクリプトにある cash のように書きます。個数に制限はありません。変数内の番地 (index) が i であるデータ呼び出すためには cash[i] と書きます。各データの番地はゼロから始まります。なので、cash[0]=-4.0、cash[1]=1.0、cash[5]=0.9 となっています。

```
*****  
# computing of the Present value of the project  
cash=[-4.0, 1.0, 1.2, 1.3, 1.1, 0.9] #単位は億円  
r=0.05  
pv=cash[0]+cash[1]/(1+r)+cash[2]/(1+r)**2+cash[3]/(1+r)**3+cash[4]/(1+r)**4+cash[5]/(1+r)**5  
print('the present value of the project = ', pv)
```

このファイルを example.py という名前で保存します。保存の仕方は各アプリで異なります。このファイルを実行すると、0.7739...と結果が表示されます。プロンプト上での入力と異なり、結果を表示させるためには、print 文を書く必要があります。

初期の投資資金を含めて計算しているので、ファイナンスでは、これを正味現在価値 (net present value) と言います。割引率が異なる時、あるいは、キャッシュフローが変化するとき、このファイルの対応する箇所の値だけを変えて、ファイルの文を部分的に修正して実行します。例えば、今後3年間に渡り、年120万円のキャッシュフローを生み出す投資プロジェクトの現在価値を計算したいときは、変数 cash の部分だけを

```
cash=[0,120,120,120,0,0]
```

と修正すれば良い。

任意の期間のキャッシュフローに対応する計算をするためには、条件文 for を活用する必要があります。条件文について若干説明します。整数1から9までの総和を求めるプログラムは

```
*****
```

```
wa=0  
for i in range(10):  
    wa=wa+i  
print(wa)
```

```
*****
```

と書きます。range(10) は整数0から9までの値の集合を表現します。つまり、range(10)=[0,1,2,3,...,9] です。

```
i in range(10)
```

は、i が整数 [0,1,2,3,...,9] のどれかに一致することを表現します。

```
*****
```

```
for i in range(10):
```

```
*****
```

は、i が0から9までの整数のいずれかに一致する限り、インデント (字下げ) を持つラインの命令文を繰り返し実行しなさいという条件文です。for 条件の次のラインは必ずインデント (字下げ) を持つ文となります。インデントを持つ文は wa=wa+i です。条件が満たされなくなると、繰り返しが終了し、この条

件文から抜け出します。このプログラムがどのように作動するのを見ましょう。i=0 から始まりますので、wa=wa+0 が実行されます。初期条件で wa=0 と設定されていたので、ここでは、まだ wa=0+0=0 となっています。この for 文では、i の値は整数で順番に大きくなります。i=1 になると、wa=wa+1 が実行されます。wa=1 になっています。i=2 になると、wa=wa+2 が実行されます。wa=3 になります。i=3 になると、wa=wa+3 が実行されます。wa=6 になります。i の値がだんだん大きくなって、i=10 になると、[0,1,2,3,...,9] の範囲を越えるので、条件を満たさなくなります。そこで、for 条件文からぬけ出して、次のラインに行きます。つまり、print(wa) が実行されます。45 と表示されます。皆さんも各自で実証してみてください。

以下のようなスクリプトを作成して下さい。保存して、実行して下さい。

```
# computing of the Present value of the project
cash=[-4.0, 1.0, 1.2, 1.3, 1.1, 0.9] #単位は億円
r=0.05
pv=0.0
for t in range(len(cash)):
    pv=pv + cash[t]/(1+r)**t
print('the present value of the project = ', pv)
```

'the present value of the project = 0.7739...' と表示されれば成功です。ここで、len(cash) は cash のデータの個数（長さ、length）を意味する。この例では、len(cash)=6 です。range(len(cash)) は一般的に整数の区間 [0,1, 2, ..., len(cash)-1] を表現しています。なので、

```
range(len(cash))=[0,1,2,3,4,5]
```

となっています。t の値がこの整数値の区間内にある限り、

```
pv=pv + cash[t]/(1+r)**t
```

を繰り返します。上記の条件文は、キャッシュフローの期間を限定していない一般的なものなので、いかなる長さのキャッシュフローに対しても適用可能です。キャッシュフローのデータ (cash) と割引率 (r) を与えるとき、現在価値計算の結果を返してくれるスクリプトファイルになっています。