

2. 簡単な計算 : Getting started through simple calculations

プログラミング・データサイエンス I

2021/4/22

1 今日の目標

今日の目標

- プログラミング環境に慣れる
- プログラムで使う基本演算を知る

今日は、プログラミング環境に慣れることが一番の目標です。実際にプログラムを少し書き、実行しましょう。

プログラミング言語は、「言語」ですから、始めはわからないことが出てきても気にする必要はありません。今日は、基本的な演算を行っていきます。それ以外のことも、少し出てきますが、「だいたい」のイメージがつかめれば十分です。

1.1 サンプルプログラムの取得

サンプルプログラムの取得

- Github からサンプルプログラムを取得
- ダウンロードしたファイルを確認

始めに、今回使用するサンプルプログラムをダウンロードしましょう。VSCode を起動します。

左下の歯車のアイコンを押し、一番上の「コマンドパレット」を選びます。現れたサブウィンドウの検索窓に「Git」と入れると、「Git クローン」があります。それを選ぶと、リポジトリ名を聞いてきますから、以下を入力します。

```
https://github.com/first-programming-saga/fundamentals
```

保存先を聞いてきますから、作業用フォルダ PythonProject を指定します。

作業用フォルダ PythonProject の下に、fundamentals というフォルダが出来ています。その下に、ファイルが出来ました。

2 プログラムの書き方

2.1 プログラムの書き方:1

プログラムの書き方:1

- フォルダの開き方
- Cell の中にプログラムを書く
 - Cell 毎に実行できる
 - 全てのセルを上から順に実行することもできる
- Cell Type を Markdown とすると、テキストとなり、説明に使える
- プログラム中のコメントは”#”で開始

まずは、先週の復習からはじめましょう。前回は、簡単なプログラムをセルに書き、セル毎に実行しました。何がセルだったかは、後でもう一度説明します。全てのセルを上から下まで、順に実行することもできます。後でやってみましょう。

それでは、サンプルプログラムを開きます。先程ダウンロードした fundamentals というフォルダにマウスを合わせて右クリックします。メニューの中にある「Code で開く」で開きます。左側には、フォルダ内のファイル一覧が見えます。

今回サンプルプログラムでは、セルの外側に日本語で説明が書かれている部分があります。このように、セルのタイプを” Markdown ”と設定することで、説明を追加することができます。このテキスト部分は、プログラムとしては実行しません。

プログラム中に” # ”を書くと、そこから行末までがコメントとなります。つまりプログラムとして実行しない部分となります。

2.2 プログラムの書き方:2

プログラムの書き方:2

- 文の区切り
 - 改行か” ;” (セミコロン)
- 行途中での折り返し
 - バックスラッシュ”\”を使って、継続していることを示す。
”¥”でも同じ
- 大文字と小文字は区別
- 変数、演算子、関数の間のスペースは見やすいように適当に入れてよい
- 行の先頭のスペースは意味がある

プログラムの書式をまとめておきます。後で、例を見ながら確認すれば十分です。

一つの文は、改行で区切るのが基本です。どうしても、一行に複数の文を書きたい場合には、セミコロンで区切ります。

一行が長くなってしまう場合には、適当なところで折り返すと見やすくなります。その場合には、バックスラッシュ”\”を使います。日本語キーボードでは、円マーク”¥”です。

プログラムを書く上で重要なことの一つに、大文字と小文字の区別があります。別の文字として扱われます。注意してください。

変数、演算子、関数の間には、見やすいように適当にスペースを入れます。ただし、行の先頭のスペースには意味があります。5回目以降に説明します。

3 計算と代入

計算と代入

- 基本的な演算記号
- イコール記号の意味

最初に演算記号の意味を見ていきます (表 1)。四則演算の記号は大丈夫でしょうか。掛け算が” *” となっています。キーボードに「×」が無いですからね。

最後の三つは少し説明が必要ですね。a//b は、a を b で割ったときの整数部分を返します。例えば、1//0.3 を実行すると 3.0 となります。a%b は、a を b で割った時の余り

演算子	例	説明
+	$a + b$	加算 $a + b$
-	$a - b$	減算 $a - b$
*	$a * b$	乗算 $a \times b$
/	a / b	除算 a/b
//	$a // b$	a を b で除した整数部分 $[a/b]$
%	$a \% b$	a を b で除した余り $a \bmod b$
**	$a ** b$	a を b 回乗する a^b

表1 基本的演算

を返します。1%0.3 は、0.1 を返します。本当のことを言うと、ちょうど 0.1 とはなりません。後で説明します。a**b は、a を b 回かけたものです。b は整数である必要はありません。

先ほどの、1%0.3 がちょうど 0.1 にならない理由です。コンピュータの中は、2 進数で動いています。整数は良いのですが、問題は小数です。1/2 や 1/4 のような、2 のべき乗の逆数は正確に表現できます。しかし、0.3 は正確に表現できず、有効桁の一番最後にちょっと誤差が入っています。これが理由です。ここは、ちゃんと理解できなくても大丈夫です。

もっとも注意が必要なのは、” = “です。これは、両辺が「等しい」という記号ではありません。この記号は、右辺を計算した結果を、左辺に代入する記号です。「等しい」という記号は、後で出てきます。

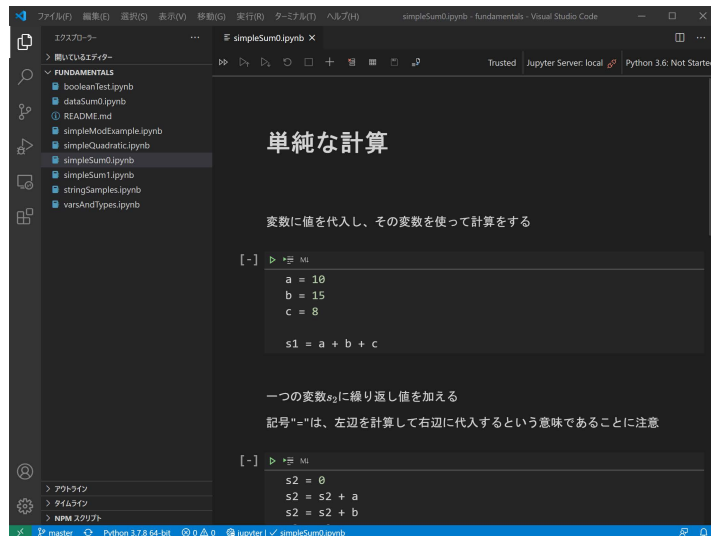
4 例 : simpleSum0.ipynb

基本的な計算

- 基本的な計算
- simpleSum0.ipynb
- =の記号の意味

始めに、simpleSum0.ipynb を開けてみましょう。初めて開くファイルの場合、そのファイルを信用してよいかというメッセージが右下に出ます。「Trust」ボタンを押してください。

図1 simpleSum0.ipynb



プログラムはセルという単位になっています。セルは、入力できる部分です。

最初のセルでは、a、b、c という三つの変数に値を代入しています。これらを加えた値を s1 という変数に代入します。セルの左上にある”▶”を使って実行します。このボタンを「Run ボタン」と呼ぶことにしましょう。

を押して実行しても何も表示されません。値は、s1 に保存されてしまったからです。

```
1 a = 10
2 b = 15
3 c = 8
4
5 s1 = a + b + c
```

二番目のセルでは、s2 という変数に 0 を代入し、順次、a、b、c という三つの変数を加えてきます。この例はとても重要です。記号”=”は代入の意味であったことを思い出してください。一つの変数 s2 に、次々と値を加えていることで、s2 という変数の値が変わっていきます。このセルの最後に、s1 と s2 を print コマンドを使って印刷します。「Run」ボタンで実行してみましょう。同じ値が印刷されますね。

```
1 s2 = 0
2 s2 = s2 + a
3 s2 = s2 + b
4 s2 = s2 + c
5
6 print(s1,s2)
```

最後に二つ課題があります。やってみましょう。エラーが出ても気にしなくて大丈夫です。まずは、一人で悩みましょう。次は、友人に聞きましょう。出来た人は、友人に教えてください。どうしても分からないときには、私に聞いてください。

全体を実行する必要がある時には、メニューの一番左の”▶” 「Run All」を選びます。

課題 1 上記の a 、 b 、及び c を使って以下のような式を計算しなさい。

$$x = a * c \quad (4.1)$$

$$y = b/c \quad (4.2)$$

$$z = (a + b) * c \quad (4.3)$$

課題 2 演算 $b//c$ と $b\%c$ を実行し、その結果を観察しなさい。

5 例 : simpleQuadratic.ipynb

例 : simpleQuadratic.ipynb

- 二次方程式の解
- 判別式を使った場合分け
- 複素数

最後の例は、二次方程式の例です。 a 、 b 、 c を実数とします。二次方程式

$$ax^2 + bx + c = 0 \quad (5.1)$$

の解を求めましょう。判別式

$$D = b^2 - 4ac \quad (5.2)$$

が負でないとき、解は

$$x_{\pm} = \frac{-b \pm \sqrt{D}}{2a} \quad (5.3)$$

ですね。特に $D = 0$ のときは、解は一つです。 $D < 0$ の時は、解は複素数となり

$$x_{\pm} = \frac{-b \pm i\sqrt{-D}}{2a} \quad (5.4)$$

です。ここで i は虚数単位 ($i^2 = -1$) です。

それでは、プログラムを見ていきます。最初のセルに、`import math`とあるのは、`math`というライブラリを使うことを示しています。今回は、平方根を求めるために使用します。ライブラリの使い方は、4回目で説明します。

二番目のセルで、三つの係数 a 、 b 、 c を定め、三番目のセルで判別式を計算しています。判別式の値に応じて、4番のセルで解を求めています (ソースコード 5.1)。`if` が条件に応じた処理をする命令です。詳しくは5回目以降で説明しますが、だいたいの意味は理解できるでしょう。なお、`1j` は $1 \times i$ を表しています。

ソースコード 5.1 二次方程式の解

```
1 if d >= 0: #実数解
2     x_1 = (-b + math.sqrt(d)) / (2 * a)
3     x_2 = (-b - math.sqrt(d)) / (2 * a)
4 else: #複素数解
5     x_1 = (-b + 1j * math.sqrt(-d)) / (2 * a)
6     x_2 = (-b - 1j * math.sqrt(-d)) / (2 * a)
7 print(x_1, x_2)
```

課題 3 a 、 b 、及び c を適当に変化させ、解が分かる例題で確かめなさい。例題は、二つの実数解となる場合、重複解となる場合、複素共役解となる場合を作りなさい。

6 次回

次回は、変数の使い方を学びます。教科書では、3章「値と変数」です。