

情報処理概論

02. 変数と型 / 算術計算



” Hello, world” (文字列) や 5 (整数) などの具体的なデータをオブジェクトと言う.

文字列

'A', 'a' ...

半角シングルクォート

"Hello, world" ...

半角ダブルクォート

整数

5 2

浮動小数点数

5.0 5. 0.2 .2 ...

- オブジェクトの**型**: オブジェクトの種類を表す
- 例:

- **int**: 整数型 (10, -3, ...)
- **float**: 浮動小数点(実数)型 (3.14, -2.0)
- **str**: 文字列型 (“おはよう”)
- **list**: リスト (次回以降で説明) ([1, 3, 4, 6]) 等

本日は特にこれらを説明

変数 … オブジェクトを入れるもの（箱）

変数

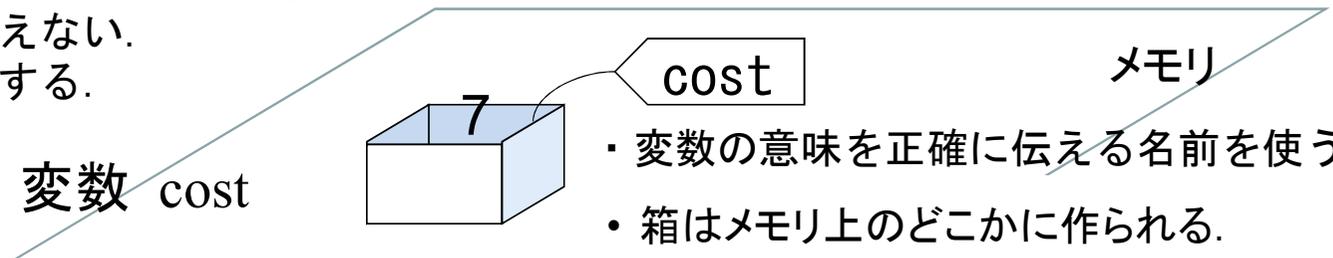
オブジェクト

例： $A = 7$

変数名

- 英数字かアンダーバー（`_`）で作られる。
- 数字は最初の文字に使えない。
- 大文字と小文字を区別する。

例：変数 `cost`



変数への代入

- 変数へオブジェクトを“格納”すること
- Python言語ではイコール(=)を使用するが、左辺と右辺が等しいという意味ではない。

例： $A = 7 + 8$

左辺はオブジェクトを格納する変数を1つだけ書く

右辺はオブジェクトまたは計算式

例： $cost = cost + 1$

更新された新しい値が代入される

更新される前の古い値が計算に用いられる

変数の値を画面へ出力する： `print(変数)`

例： `print(cost)`

代数演算子

+a	正数
-a	負数
a+b	加算
a-b	減算
a/b	除算
a%b	aをbで割った余り (剰余)
a**b	aのb乗
a//b	切り捨て除算

演算される変数がともに整数型のときの答えは整数型, それ以外のときの答えは浮動小数点型 (実数型) となる.

代入演算子

a=b	aにbを代入する
a += b	a = a + b に同じ
a -= b	a = a - b に同じ
a *= b	a = a * b に同じ
a /= b	a = a / b に同じ
a %= b	a = a % b に同じ
a **= b	a = a ** b に同じ
a //= b	a = a // b に同じ

```
X = 10
Y = X / 3
print(Y)
```

```
X = 10.0
Y = X / 3.0
print(Y)
```

実行結果
3

実行結果
3.3333333333



Pythonではこうはならないが、VBAではこうなるので注意

算術関数

数学的な関数はライブラリ関数（モジュール）として用意されている（算術関数モジュール：math）。

- import math で以下のような関数を使用できる。
- 関数 sin(), cos(), tan(), log() など
- 括弧の中に引数を書く。
- 使えるライブラリ関数はできるだけ使う。

モジュールの使い方

1. import モジュール名

- モジュール内の関数を使えるようにする。モジュール名.関数名で関数を指定。

2. import モジュール名 as 略称

- 1 と同様だが、モジュール名を略称で置き換えることができる。

3. from モジュール名 import 関数名

- モジュールから指定した関数のみを使えるようにする。モジュール名は指定不要。

使用例:

使用するモジュール

関数名

```
import math
x = r * math.cos(theta)
```

引数:

- 関数に与えるパラメータ
- 変数, 式などを書くことができる

```
import math
x=math.cos(0)
```

```
import math as m
x=m.cos(0)
```

```
from math import cos
x=cos(0)
```

mathで定義されている関数の例

返り値の型	関数名	説明
float	cos (float x)	余弦(cos)を計算
float	sin (float x)	正弦(sin)を計算
float	tan (float x)	正接(tan)を計算
int	ceil (float x)	x以上の整数で最小のもの
int	floor(float x)	x以下の整数で最大のもの
float	sqrt(float)	xが0以上なら平方根を計算
int	gcd(int x, int y)	xとyの最大公約数

mathには、他にもたくさんの関数が定義されている

<http://docs.python.jp/3.6/library/math.html?highlight=math#module-math>

算術計算のサンプルプログラム

Log(1+x) ただし (-1<x<1)のマクローリン展開

$$\log(1+x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \dots$$

上記のマクローリン展開近似を確認するプログラム

【注意】
日本語を使う場合は、ファイルの
文字コードをUTF8にすること。

```
import math
print("xの値を入力してください ただし-1<x<1")
Moji = input()
x = float( moji )
f1 = x-(1/2)*(x**2)+(1/3)*(x**3)-(1/4)*(x**4)
f2 = math.log(1+x)
print("マクローリン展開近似したlog(1+x)=", f1)
print("math.log(x)の計算結果=", f2)
```

半角文字

半角文字

キーボードから文字列入力し変数mojiへ格納

文字列をfloat(浮動小数)型へ変換し、変数xへ格納

計算の優先順位を明確にするため()カッコで式を表現

半角文字

まとめ

オブジェクトの型: 整数型 / 浮動小数点型

変数、変数への代入、変数の画面出力 print関数

代数演算子、算術演算関数、mathモジュール

第2回 レポート課題

$\text{Log}(1+x)$ のマクローリン展開の確認プログラムに修正を加え、

$0 < x < 2\pi$ の範囲で $\cos(x)$ をテイラー展開によって多項式近似する式と、

`math.cos(x)` で計算した結果を比較するプログラムを作成せよ。

ただし、上記の全ての x の範囲で `math.cos(x)` との差が 0.0001 以下になる十分な高次近似式のコードを書くこと。

下記の課題提出用フォルダへ、課題の番号と提出者が分かるようにファイル名を以下のようにしてアップロードせよ

第2回1TE19xxxZ名前.py

https://share.iii.kyushu-u.ac.jp/public/IRbwAAVITI5A2X4BE45t6TqQIE0UQSQUI5Bap_kZ_sjy

講義資料、および上記フォルダへのリンクは下記ホームページから

<http://sysplan.nams.kyushu-u.ac.jp/gen/edu/InfoProcess/2019/index.html>