

第 1 回 Python 講座

2018 年 7 月 6 日

1 Python 基本知識

1.1 特徴

1.1.1 可読性

python は一般的に他のプログラミング言語と比べて可読性が高いとされています。それは python においてインデントが重要であるからです。他の言語ではインデントは個人の趣向によって決められることが多いのですが python では if,for から関数定義まですべてがインデントによって管理されます。よって誰が書いてもインデントがきちりしているためプログラム全体が見えやすいです。またより少ないコードでプログラムが書けるように工夫されています。小規模から大規模までわかりやすいように配慮されています。

1.1.2 ライブラリ

python ではそのライブラリの豊富さがよく言われています。膨大なライブラリがインターネット上で公開されておりパッケージ管理ツールによって手軽にライブラリをプログラムに追加できるようになっています。python ではこれにより、web アプリケーションやバッチ処理、デスクトップアプリケーションの開発、さらには科学計算、機械学習など様々な場面で用いられています。また後期に開催予定である機械学習講座ではこのライブラリを用います。Tensorflow 等様々なものがありますが私の都合上 Chainer を用いて行おうと考えています。

1.1.3 対話型でありオブジェクト指向

python は一般的に対話型言語であるといわれています。しかしその一方でオブジェクト指向言語でもあります。よって python では扱うデータがすべてオブジェクトとなっています。しかし python ではオブジェクト指向言語であることをそこまで意識しなくても大丈夫です。これも python の特徴でもあります。

1.2 プログラミングでの基本事項

1.2.1 インデント

先ほども記述しましたが python ではインデントによって処理が区別されています。C や Java では {} でブロックを構成していましたが python ではそれがインデントになります。

1.2.2 コメント

python でコメントを書きたい場合は#もしくは""" を用います。1行のみの場合#を複数行をコメントにしたい場合"""で囲むことで出来ます。また文字コードが違うため何も指定な場合は日本語でコメントすることは止めましょう。もし日本語でコメントしたい場合は

日本語コメント

```
#coding: utf-8
print("Hello world") #Hello world を出力する
```

このように文字コードを変更しましょう。またもしかしたらエディタによっては自動でやってくれる場合もあります。なお python では C 言語のように {} で囲むのではなくインデントで管理されていることに気を付けてください。

2 入出力

ここからは実際に python を使って様々なプログラミングをしてもらいます。まずは入出力関係です。

2.1 出力

python における出力は print() 関数を用います。

print

```
c=18
print(" 文字入力")
print(" a+b=",c)
```

コマンドライン

```
文字入力
a+b=18
```

このようにダブルクォーテーションで囲んだ部分はそのまま、それ以外のところで変数かつ数値等が入っている場合は変数の値が出力されます。また出力の際、変数を複数扱うのに整理しやすいように format() 関数が用意されています。format() 関数は出力における変数を一括管理する関数となります。

format

```
a,b,c,d=1,3,5,7
print(" a={0},b={1},c={2},d={3}".format(a,b,c,d))
```

コマンドライン

```
a=1,b=3,c=5,d=7
```

2.2 入力

2.2.1 型

python における入力は `input()` を用います。しかしこのとき型に注意しなければいけません。C や Java では整数型 (`int, short, long` など)、浮動小数点数型 (`float, double`) でしたが python では整数型 (`int`)、浮動小数点数型 (`float`) です。以下の表に型についてまとめます

型	値	例
<code>int</code>	整数	1,2,10
<code>float</code>	浮動小数点数	1.2 33.4,0.07
<code>bool</code>	論理値	True or False
<code>complex</code>	虚数	3.14j,2i
<code>str</code>	文字列	"Hello"

型について調べたいときは `type()` 関数で調べることができます。

————— `type()` —————

```
print(type(10))
print(type("Hello"))
```

————— コマンドライン —————

```
< class'int' >
< class'str' >
```

2.2.2 動的型付け

python では入力の特徴として動的型付けがあります。これはや Java のように変数宣言時に肩を明示する必要がありません。初期化等を行う時も変数宣言しなくてよいです。変数の方は値から自動で判定してくれるためあまり気にしなくても構いません。しかし、入力の際には型を気にしなくてはエラーを出されてしまいます。それは `input()` が初期値では `str` 型に設定されているためです。ですので数値の入力等を行う差には明示的に宣言しなくてははいけません。

————— `input()` —————

```
str 型の読み込み
n=input()
int 型の読み込み m=int(input())
```

3 条件分岐

python における条件分岐は `if` 文にて行います。if 文はインデントによって管理されます。

```
if 条件 1:  
    処理 1  
elif 条件 2:  
    処理 2  
else:  
    処理 3
```

また C や Java と違って python には switch 文がありません。似たような動作を行わせたい差には if,elif,else で代用して記述するようにしてください。

4 演習課題

1. 好きな文字列を入力し、その文字列の出力及び型出力せよ
2. Fizzbuzz 問題に対し、好きな数値を入力したときに Fizz or Buzz or FizzBuzz を出力せよ