

Python勉強会

4歩目
(実質1.5歩目)

チャンスラボ株式会社 岡本



CoderDojo太宰府

0. はじめに。

今回は ネットに繋いで情報取得するプログラム作りますので、
以下のWi-Fi設定をしてください。

Python_Lab

20191205

1. 基本構文など

これまで、言語予約語、変数、など端折ってきましたが。

受講者よりリクエスト有ったので基本に立ち返ります。

どのプログラミング言語でも、オリジナル関係の資料は大元のドキュメント見るのが基本です。

<https://docs.python.org/ja/3/library/functions.html>

今回は基本的な使用例を説明するので、応用例などは各自調べてください。

全部教えて貰わないと分からないよ！とか思ってる人居ますか？

私も含めて大半のベテランでも全部の仕様なんて把握していません。

ベテランのプログラマでもパーフェクト超人ではありません。

むしろ社会的には・・・

★解らないときは周りに聞いてください。仲魔が増えます。

2. 組み込み関数

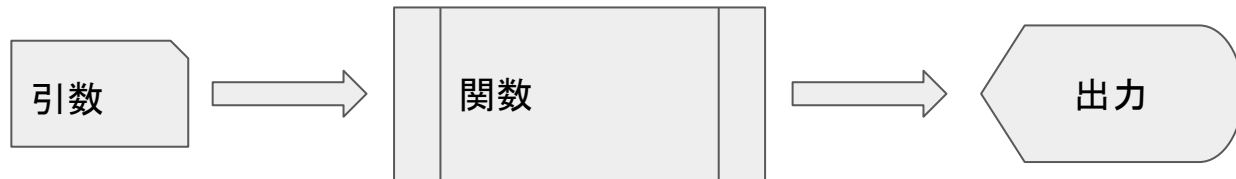
Python には多くの関数と型が組み込まれています。

今回はその中の一部を説明します。

関数とは、数学の関数の様に、「何らかの値を渡すと、その値に応じた何らかの値を返すもの」です。

実際にプログラミング言語で使う場合は**引数**を渡して**関数**を実行すると**結果**を得る事が出来ます。

結果は計算結果を**戻り値**を返えしてきたり、画面等に**出力**したりします。



3. print()関数

とりあえず、1歩目でも紹介した`print()`関数です。

()内の物を出力します。

通常は画面に出力しますが、出力先をファイル等に指定することも可能です。

文字列を出力する。

```
print("Hello world!")
```

計算を行う。

```
print(1+2)
```

実行結果は以下のようになります。



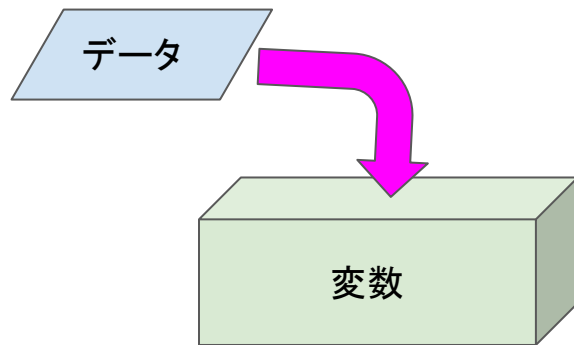
```
cmd コマンド プロンプト - python
>>>
>>> print("Hello world!")
Hello world!
>>>
>>> print(1+2)
3
>>> _
```

4. 変数

変数とは、数値などのデータを保持する場所です。

後からそのデータを利用することができます。

プログラムをする際は、大抵は使用します。



PyCharmのPythonコンソール又はコマンドラインで以下を実行してください。

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916
```

```
>>> PI = 3.14
```

```
>>> r = 4
```

```
>>> ANS = 2 * PI * r
```

```
>>> print( ANS )
```

```
25.12
```

```
>>>
```

PI と r と ANS が変数です。

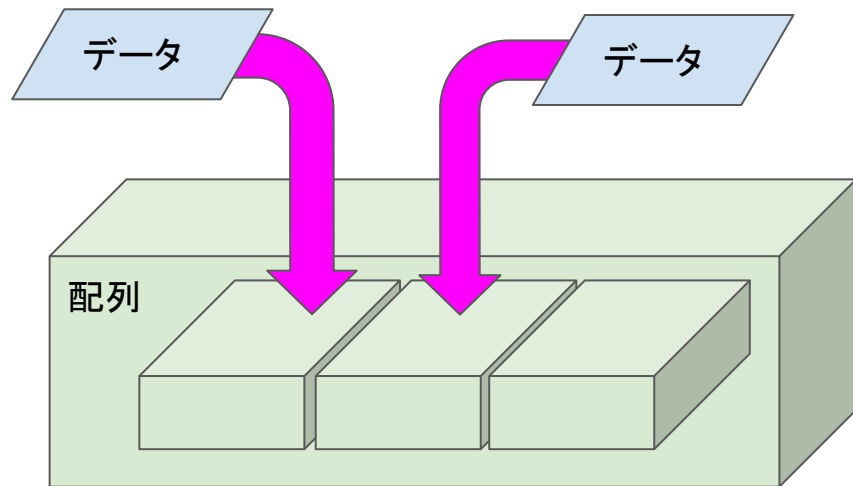
5-1. 配列

配列とは、複数の値まとめて入れる事の出来る変数のことで、
複数の箱が並んでいるイメージと認識して良いです。

Pythonには配列の様に複数值を
まとめて持つ形式が幾つかあります。

- ・リスト型
- ・タプル型
- ・辞書型
- ・配列型

※当資料での説明ではリスト型なども配列として説明します。



5-2. リスト型の使用

今回はPythonで一般的なリスト型を使用します。

PyCharmのPythonコンソール又はコマンドラインで以下を実行してください。

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916
```

```
>>> data_list=[1,3,5,7,11]
```

`data_list` が リスト型 の変数です。

```
>>> print(data_list[3])
```

```
7
```

リストの中から使用したい位置を指定します。
※位置指定の初っ端は「1」ではなく「0」

```
>>> data_list[3]=9
```

```
>>> print(data_list[3])
```

```
9
```

リストは内容の変更も可能。
※タプルは変更駄目

6. 条件分岐

条件を満たすか否かで次に行う処理を変えるものです。

条件を満たす場合を **真**(true)、満たさない場合を **偽**(false) と言います。

正にイラストの様な処理と成ります。 ※ルールに従い動くだけ。

以下を実行してください。

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492,
```

```
>>> train = "blue"
```

```
>>> if train == "red":  
...     print("pull")  
...
```

train が blue の時は何も
処理しない。

```
>>> train = "red"
```

```
>>> if train == "red":  
...     print("pull")  
...
```

train が red の時は
pull を出力する。

```
pull
```



7. 繰り返し処理

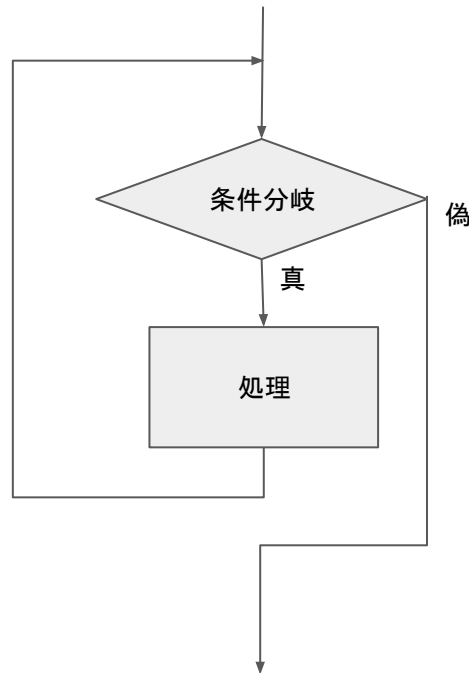
繰り返し処理とは同じ処理を何回もやらせることで、**ループ処理**とも言います。

Pythonでは **while** もしくは **for** が利用できます。

今回は **while**文を使用します。

以下を実行してください。

```
>>> n=0
>>> while n<3:
...     print("nは" + str(n))
...     n = n + 1
...     print("end")
nは0
nは1
nは2
end
```



8. 演習問題

下記のリスト内の数値を小～大へ並び変えて画面出力しなさい。

```
data_list=[10,1,5,2,7,2,0]
```

今回の勉強会で学習した **if文**、**while文**を使用して実装すること。

出力結果は以下の様に成る事。

```
0,1,2,2,5,7,10
```

注意点:

- ・冗長な実装をしない事。
- ・他人が読み易く実装する事。
- ・実装内容を他人に説明できること

9. 補足 ソート

「8.演習問題」の様に、値を順に並べる頃を**ソート**と言います。

ソートには色々な手法があります。

以下は一例

- ・バブルソート
- ・ヒープソート
- ・クイックソート

etc

↓ ソートを分かり易くアニメ風に可視化した実例 (C#)

<https://qiita.com/r-ngtm/items/f4fa55c77459f63a5228>

↓ Pythonの参考実装例

<https://github.com/CoderDojoDazaifu/Works/tree/master/Python/Sort>

今回使用した資料について

今回使用した資料等は、以下のgithubに置いていきますので
ダウンロードして使用可能です。

https://github.com/okamotomasatosi/py_doc

