

# Python講座

2016/4/28(木)

# 今回の内容

- ▶ 変数
- ▶ input()
- ▶ 演算子と式
- ▶ 複合代入演算子
- ▶ リスト
- ▶ リストの切り出し
- ▶ 数字のみのリスト
- ▶ リストに関する文
- ▶ 辞書
- ▶ 比較演算子
- ▶ if文(条件文)
- ▶ if文の派生
- ▶ 複数の条件式
- ▶ for文(ループ文)①
- ▶ for文②
- ▶ while(ループ文)
- ▶ ループ文に関する文
- ▶ ループ文に関する文(例)
- ▶ 演習問題

# 変数

- ▶ 変数 a A hensu
- ▶ 変数の表示 print(変数)  
print(a) print("{}".format(hensu))
- ▶ 変数への入力 変数=代入したい値 input()  
a=10 A="moji" a=input()
- ▶ **文字自体**を扱う '文字列' "mojiretsu"

# input()

- ▶ input()はキーボードからの入力を受け付ける文
- ▶ 入力を受けたものは文字列(" ")として扱われる
- ▶ 数字として扱いたい場合は

```
a=int(input())
```

という形で、int()で囲む

- ▶ ()ではprint()ができ、入力直前に表示される

# 演算子と式

- ▶ 加算  $a+b$  減算  $a-b$  乗算  $a*b$  除算  $a/b$
- ▶ べき乗を求める  $10^{**}3 >>>1000$  (10の3乗)
- ▶ あまりを求める  $14\%3 >>>2$  (4あまり2)
- ▶ 少数以下の切り捨て  $10//4 >>>2$  (2.5の切り捨て)

※計算順序は数学と同じ、**べき乗**は乗算より**先**

# 複合代入演算子

- ▶ 変数に計算結果を代入する際、以下のように省略しながら書くことが可能

変数=変数 演算子 値 → 変数演算子=値

power=power+3 → power+=3

powerに+3した値をpowerに代入する

※変数に値が代入されている状態でのみ可能である

# リスト

- ▶ リスト名=[要素,要素,⋯]

```
list=[1, 2, 3, 4] A=["文字", "moji"]  
mix=[1, "moji", 3]
```

- ▶ リストの要素を**変数**として扱う

```
変数=リスト名[要素の番号]
```

※要素の番号の最初は0である

```
print(mix[1])
```

```
>>>moji
```

# リストの切り出し

- ▶ リスト名[番号:番号]
- ▶ `sentence = ['The', 'quick', 'brown', 'fox']`  
`print(sentence[2:4])`  
`>>> ['brown', 'fox']`
- ▶ `adjectives = sentence[1:3]`  
`print(adjectives[0])`  
`>>> quick`



# 数字のみのリスト

▶ `sum`(リスト名)    `min`(リスト名)  
   `max`(リスト名)

▶ `s = [12,3,456,7,89]`  
   `print ("{} , {} , {}".format(sum(s),min(s),max(s)))`  
   `>>> 567 , 3 , 456`

# リストに関する文

▶ len(**リスト名**)

リストの**要素数**を出力する

▶ eval("[**要素,要素,...**]")

" "の中でリストを書くことで、**リストとして**使用することができる

▶ a=eval("[1,12,123,1234]")

```
print (len(a))
```

```
>>> 4
```

# 辞書

- ▶ リストの要素に対して名前(数字でも可)を付けることで、番号ではなく名前呼び出す
- ▶ 辞書名 = {名前:要素, 名前:要素, ...}
- ▶ MIX = {"RPG": "FF", "卯月": 4, 334: "阪神", 33: 2}

```
print(MIX[334])           >>> 阪神
```

```
print(MIX["卯月"] * 2)   >>> 8
```

# 比較演算子

▶ 以降のページに出てくる**条件式**に対して使用する

Pythonでの書き方	数学での書き方	意味
<code>a == b</code>	$a = b$	aはbと等しい
<code>a != b</code>	$a \neq b$	aはbと等しくない
<code>a &lt; b</code>	$a < b$	aはbより小さい
<code>a &gt; b</code>	$a > b$	aはbより大きい
<code>a &lt;= b</code>	$a \leq b$	aはbより小さいか等しい(以下)
<code>a &gt;= b</code>	$a \geq b$	aはbより大きいか等しい(以上)

# if文(条件文)

- ▶ **if** 条件式: 内容  
条件式が**満たされた**場合、内容を実行する
- ▶ if a > 10:  
    print(a)  
**もし**aが10より大きい**ならば**、aをprint

# if文の派生

- ▶ if 条件式: 内容

もし条件式が満たされたとき、内容を実行

- ▶ elif 条件式: 内容

以前のif,elifの条件式が満たされなかった場合かつ  
もし条件式が満たされたとき、内容を実行

- ▶ else: 内容

以前のif,elifの条件式が満たされなかった場合  
内容を実行

# 複数の条件式

▶ if 条件式a **or** 条件式b: 内容

条件式a**または**条件式bが満たされていれば、内容を実行

▶ if 条件式a **and** 条件式b: 内容

条件式a**と**条件式bの**2つ共**が満たされていれば、内容を実行

※複数つなげることも可能

# for文(ループ文)①

- ▶ `for` 変数a in range(変数b): 内容  
変数b(整数)の数だけ、内容を実行する  
変数aの初期値は0、内容が実行されるたびに+1  
されていく →最終的には、変数a==変数b - 1
- ▶ `for x in range(5):`  
    `print(x)`  
>>>0 1 2 3 4



## for文②

- ▶ **for 変数 in リスト名:** 内容  
リストの要素数だけ、内容を実行する  
変数にはリストの要素が入り、内容が実行される  
たびに変数が更新される
- ▶ `list=[12,3,"moji",5]`  
`for x in list:`  
    `print(x)`  
`>>> 12 3 moji 5`

# while文(ループ文)

- ▶ while 条件式: 内容

条件式が満たされている間だけ、回数の限りなくずっと内容を実行し続ける

※条件式を間違えると、無限ループが起こる

- ▶ a=1

```
while a<5 :
```

```
    print(a)
```

```
    a+=2
```

```
>>>1 3
```

# ループ文に関する文

## ▶ break

ループをその場で終了させる

## ▶ continue

continue以降の内容を実行せず、ループを引き続き実行する

# ループ文に関する文(例)

```
a=0
while a<10:
    a+=1
    if a==3 or a==5:
        continue
    if a==7:
        break
    print(a)
>>> 1 2 4 6
```

# 演習問題 補足

- ▶ Pythonで乱数を使う場合、`import random`と書く
  - ※基本import ○○は最上部に書く
- ▶ `random.random()`
  - 0~1の少数がランダムに出力される
- ▶ `random.randint(値a, 値b)`
  - 値a以上、値b以下の整数をランダムに出力する
  - ※他にも`random.○○`は存在する

# 演習問題①

- ▶ テーマ：ノット30(仮)を作る
  - ▶ 準備：対戦は人間vsCPU、先攻後攻を決める
  - ▶ ルール：
    - ① 先行は1から+2した値までの数を選択し相手へ
    - ② 相手は選択された数+1から+3までの数を選択し相手へ
    - ③ 以降、互いに②を行い30を選択したほうが負け
- ※ゲームの性質上、30を言わせた方が勝ち

# 演習問題① 補足

①1,2,3

②4,5

①6,7

②8,9,10

①11,12,13

②14,15,16

①17,18

②19,20

①21,22,23

②24,25

①26

②27,28,29

①30

②の勝ち

## 演習問題②

- ▶ テーマ：Numer0nを作る
- ▶ 準備：0~9の中で、同じ数字ではない3つの数字を選び、三桁の数字(Answer)をCPUに用意させる
- ▶ ルール：
  - ①人間がキーボードから三桁の数字(同じ数字でも可)を入力する
  - ②Answerの中に、桁に関わらず数字が存在する場合はBITE、数字とその桁が正しい場合はEATと表示する
  - ③人間がAnswerを入力できたら勝利



# 演習問題② 補足

②の例 Answerが「346」のとき

065 → 1 BITE

364 → 2 BITE, 1 EAT

634 → 3 BITE

333 → 2 BITE, 1 EAT

<https://ja.wikipedia.org/wiki/Numer0n>