Python文法III

松尾・岩澤研究室
MATSUO-IWASAWA LAB UTOKYO

2024/03/19

目次

- 文法I
  - 演算
  - 変数
  - データ型
- 文法II
  - コレクション
  - List
  - Tuple
  - Dict
- 文法III
  - 条件分岐(if文)
  - 繰り返し(for文)
- 文法IV
  - 関数
  - モジュール

比較演算子

比較演算子は左右の値と評価されて真偽値を表わすものです

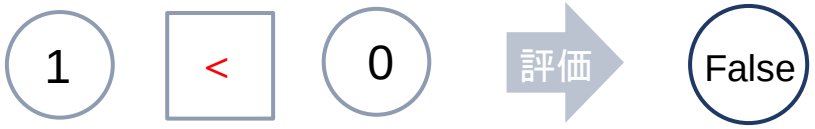
比較演算子の記号

演算子記号	意味
==	左右の値が同じ
!=	左右の値が同じでない
>	左が右の値より大きい
<	左が右の値より小さい
>=	左が右の値以上
<=	左が右の値以下

コード例

```
In  [1] : print(1 == 1)
Out [1] : True
In  [2] : print(1 < 0)
Out [2] : False
```

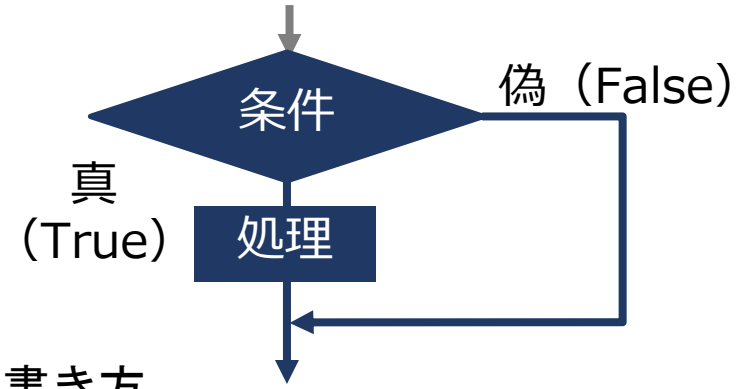
(例) 比較演算子：<（小なり）



条件分岐（if文）

条件分岐(if文)は条件の真偽値（TrueかFalse）によって**処理の流れを変更**するコードの記載方法のことを意味します

if文（もし～条件なら～する）



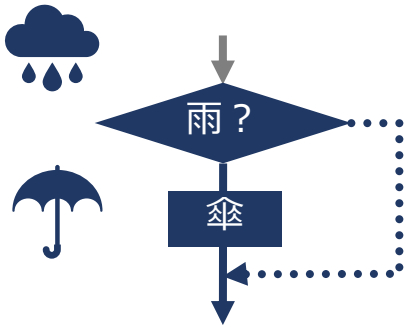
コードの書き方

```
if 条件の真偽値:
    真のときの処理
```

インデント

真偽値が真の場合のみ:（コロン）下のインデント(スペース4つ)以降の処理が実行され、偽の場合は実行されません

(例) 雨なら
傘を持つ



コード例

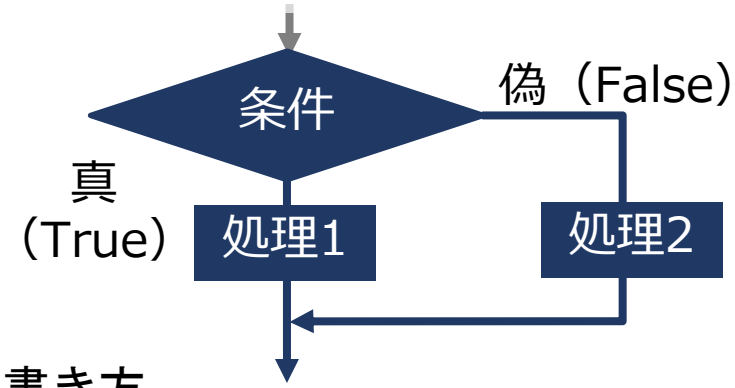
```
In  [1] :  is_rain = True
           if is_rain:
               print('傘を持つ')

Out [1] :  傘を持つ
```

※真偽値を代入する変数名には「〇〇かどうか」という意味で「is_〇〇」をよく使います

条件分岐（if-else文）

if文に**else**を使用すると、条件が偽(False)となる場合の処理を付け足せます

if-else文（もし～なら～する、
そうでないなら～する）

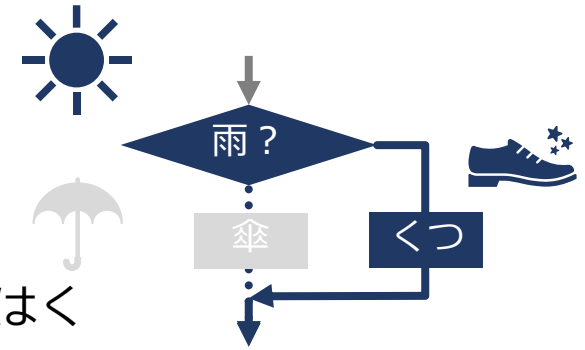


コードの書き方

```
if 条件の真偽値:
    インデント 真のときの処理1
else:
    インデント 偽のときの処理2
```

真偽値が真の場合、処理1が実行され、偽の場合、処理２が実行

(例) 雨なら
傘を持つ
そうでないなら
新しいくつをはく



コード例

```
In [1] :  is_rain = False
          if is_rain:
              print('傘を持つ')
          else:
              print('新しいくつをはく')

Out [1] :  新しいくつをはく
```

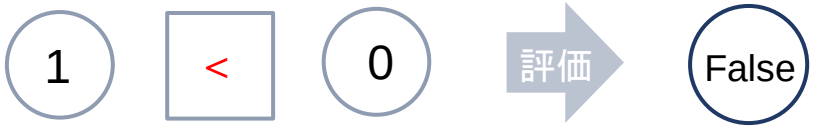
比較演算子

比較演算子は左右の値と評価されて真偽値を表わすものです。if文での条件に利用できます

比較演算子の記号

演算子記号	意味
==	左右の値が同じ
!=	左右の値が同じでない
>	左が右の値より大きい
<	左が右の値より小さい
>=	左が右の値以上
<=	左が右の値以下

(例) 比較演算子：<（小なり）



コード例

```
In  [1] : print(1 == 1)
Out [1] : True
In  [2] : print(1 < 0)
Out [2] : False
In  [3] : age = 18
          if age < 20:
              print('No Drink')
Out [3] : No Drink
```

True

論理演算子

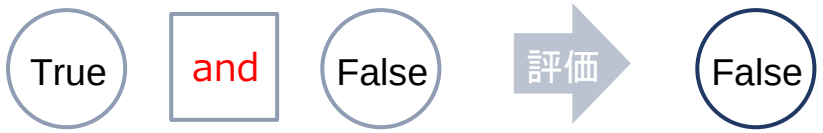
論理演算子は複数の演算子の真偽値を評価するものです
特に、if文で比較演算子と組み合わせて使用することが多いです

論理演算子の記号と優先度

演算子記号	意味
and	かつ
or	または
not	でなければ

演算子	優先度
算術	高
比較	中
論理	低

(例) and演算子



(例) not演算子

コード例

```
In  [1] : print(True and False)
Out [1] : False
In  [2] : print(True or False)
Out [2] : True
In  [3] : BMI = 20
          if BMI >= 18 and BMI < 25:
              print('normal weight')
Out [3] : normal weight
```

True and True ➤ True

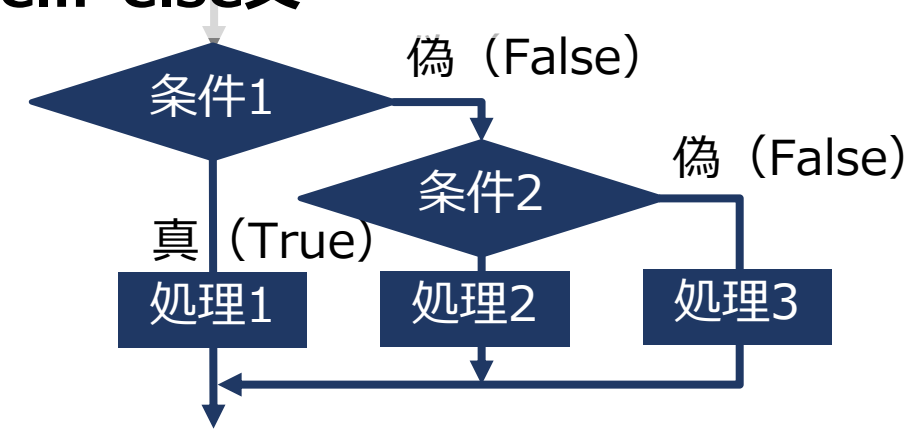
※BMIは体重と身長から算出される肥満度指標

andとorは左右の真偽値とともに評価され、notは右の真偽値のみと評価されるので注意

条件分岐（if-elif-else文）

条件分岐を3つ以上にしたい場合に**elif**を追加します
比較・論理演算子と組み合わせてより複雑な条件分岐が可能です

if-elif-else文



コードの書き方

```
if 条件1:
    インデント 条件1が真のときの処理1
elif 条件2:
    インデント 条件1が偽で条件2が真のときの処理2
else:
    インデント 条件1と2が偽のときの処理3
```

elifはelse ifの略です。elifを複数使用する場合、最初に真となる条件だけ処理が実行されます

コード例

```
In  [1] : BMI = 20

          if BMI >= 25:
              print('overweight')
          elif BMI >= 18 and BMI < 25:
              print('normal weight')
          else:
              print('underweight')

Out [1] : normal weight
```

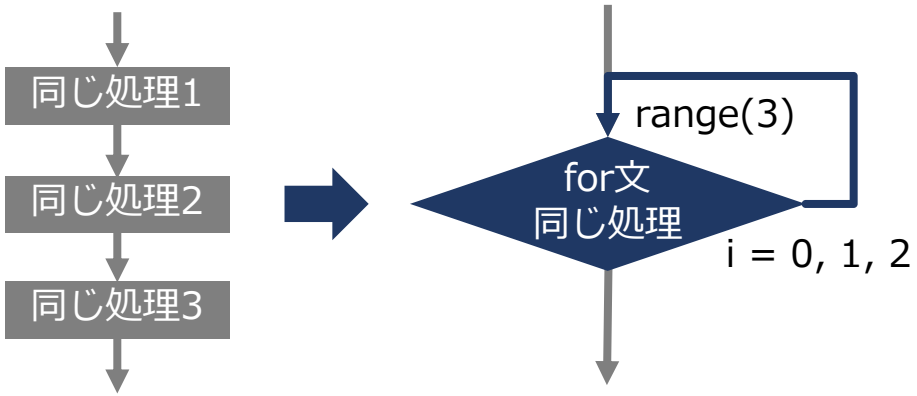
False

True

※「and BMI < 25」の部分は省略しても同じ動作

繰り返し（for文）

for文は**同じ処理を何度も書かずに繰り返したい**場合に使用します
範囲をrange()で指定して繰り返せます



コードの書き方

0, 1, 2…

```
for 変数 in range(繰り返す回数):
    繰り返し処理
```

インデント

指定回数が終わるまで:（コロン）下のインデント(スペース4つ)
以降の処理が続きます。変数は0から処理ごとに1ずつ増えます

コード例

カンマを挟むと
スペースが空く

```
In  [1] : for i in range(5) :
              print('Python文法', i)

Out [1] : Python文法 0
          Python文法 1
          Python文法 2
          Python文法 3
          Python文法 4
```

※range(5)はrange(0,5)の省略形。range(x, y)を
使用するとxから始まりy未満までと指定できます

※for文内で定義した変数に特別な意味がない数であれば、
変数名にiやjなどの文字がよく使われます

for文とリスト

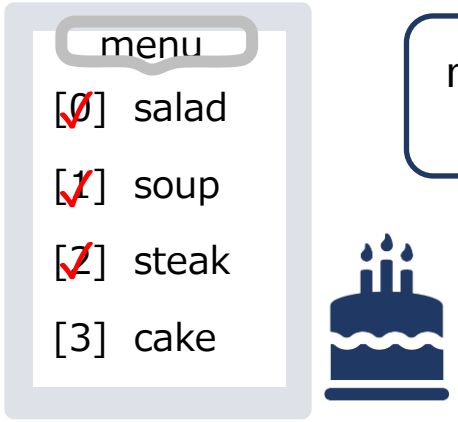
for文に**リスト**を組み合わせると**各要素を取り出し**て同じ処理を繰り返せます

コードの書き方

リスト[0]、リスト[1]、…

```
for 変数 in リスト:
インデント 繰り返し処理
```

リストの要素が変数に順に代入され、要素数分処理が繰り返されます



menu順に出される

コード例

```
In [1] : menu = ['salad', 'soup', 'steak', 'cake']

         for food in menu:
             print(food)

Out [1] : salad
          soup
          steak
          cake
```

for文とリスト

リストとfor文の組み合わせでインデックスと要素の両方を
利用したい場合、**enumerate()**が便利です

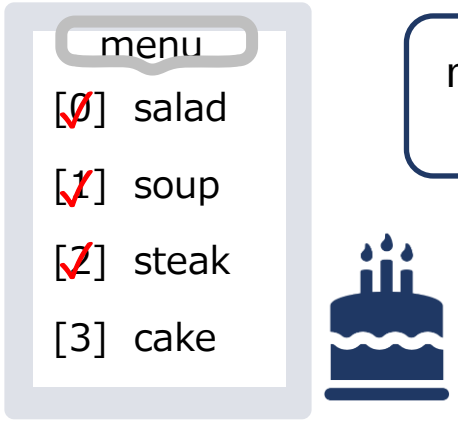
コードの書き方

0、1、…　　リスト[0]、リスト[1]、…

```
for 変数1, 変数2 in enumerate(リスト):
    繰り返し処理
```

（インデントは4スペース）

変数1にリストのインデックスが、
変数2にリストの要素が順に代入され、処理が繰り返されます



menu順に
出される

コード例

```
In  [1] : menu = ['salad', 'soup', 'steak', 'cake']

          for i, food in enumerate(menu):
              print(i, food)

Out [1] : 0 salad
          1 soup
          2 steak
          3 cake
```

enumerateは数える上げるという意味

松尾研究室

MATSUO LAB THE UNIVERSITY OF TOKYO