

## An Introduction to Python Part 4

M. Uemoto

Yoshida Lab

July 23, 2013



- ▶ 先週のおさらい
- ▶ 基本文法
  - ▶ 5章：リスト (後半)
    - ▶ スライス
    - ▶ リストメソッド
  - ▶ 出力ファイルの処理
- ▶ 演習問題

### 前回のおさらい(1)

#### リスト構造

- ▶ 大量のデータをひとつの変数に格納する
- ▶ `x = [10, 20, 30, 40, 50, 60, 70]`
  - ▶ カッコの中にカンマで区切られた値を入れる
- ▶ `x[0]`
  - ▶ リスト要素へのアクセス
  - ▶ 変数名 [添字]
  - ▶ 最初の要素 (`x[0]`), 二番目の要素 (`x[1]`) ...
- ▶ `y = [1, 2, 3] + [4, 5]`
  - ▶ 2つリストを連結して新しいリストをつくる
  - ▶ `y += [6]` リストの末尾に要素を一つ追加

### 前回のおさらい(2)

#### —— リストによるループ ——

```
color = ["Red", "Green", "White"]
for name in color:
    print name
```

- ▶ **for ループ変数 in リスト変数 :**  
リストから要素を一つずつ取り出し直下にある処理を適用
- ▶ **リスト関連関数**
  - ▶ `len(変数名)`  
リストの長さを返す
  - ▶ `min(変数名)`  
リストに含まれる要素の最小値を返す
  - ▶ 文字列変数.`split("記号")`  
特定の記号で文字列を分割する
  - ▶ `glob.glob("パス")`  
パスを含むファイル名の一覧を表示する (UNIX の `ls` に相当)

## リストのスライス

```
      0      1      2      3      4      5
data      =  ["H", "He", "Li", "Be", "B", "C"]

data[1:4]  =      ["He", "Li", "Be"]
```

- ▶ リストの一部を切りとり新しいリストを作成する
- ▶ `elem[i:j]`
  - ▶  $i$  以上  $j$  未満の添字をもつリスト要素のみを取り出す
- ▶ `elem[i:]`
  - ▶ 要素  $i$  から末尾まで
- ▶ `elem[:j]`
  - ▶ 先頭から要素  $j-1$  まで

## リストメソッド (2)

- ▶ `l.append(v)`
  - ▶ リスト  $l$  に要素  $v$  を追加
- ▶ `l.insert(i, v)`
  - ▶ リスト  $l$  の添字  $i$  の位置に要素  $v$  を挿入
- ▶ `l.remove(v)`
  - ▶ リスト  $l$  に含まれる最初の要素  $v$  を削除
- ▶ `l.reverse()`
  - ▶ リスト  $l$  を逆順に並べ替える
- ▶ `l.sort()`
  - ▶ リスト  $l$  を昇順に並べ替える
- ▶ `l.pop()`
  - ▶ リスト  $l$  の末尾の要素を取り出す

## リストのメソッド (1)

### リストメソッドの例

```
>>> color = ["Red", "Green", "Blue"]
>>> color.append("Purple")
>>> print color
["Red", "Green", "Blue", "Purple"]
>>> color.insert(1, "Yellow")
>>> print color
["Red", "Yellow", "Green", "Blue", "Purple"]
>>> color.remove("Red")
>>> print color
["Yellow", "Green", "Blue", "Purple"]
```

- ▶ メソッド
  - ▶ 変数名 + ドット + メソッド名 (パラメータ...)
  - ▶ オブジェクトに関する操作
    - ▶ 追加、削除、並べ替え

## リストによるファイル読み込み

### test.txt の読み込み

```
f = open("test.txt", "r")
for line in f:
    print line
f.close()
```

- ▶ `f = open("ファイル名", "r")`
  - ▶ ファイルを読み込みモードで開く
  - ▶ 以後、変数  $f$  はファイル (`test.txt`) を指す
- ▶ `for 変数 in ファイル :`
  - ▶ 一行ずつ文字列を読み込み、ループ変数 (`line`) に代入
  - ▶ コロン ":" をつけ忘れないこと
- ▶ `ファイル.close()`
  - ▶ 読み込み終了後は必ずファイルを閉じること

## 練習 (1)

- ▶ 今回配布した sample.tar.gz を展開
- ▶ サブディレクトリ sample1/に出力ファイルのサンプル file00.kkr から file99.kkr があります

file00.kkr

```
lattice constant
bravais=bcc  a= 5.0000  c/a= 1.0000  b/a= 1.0000
alpha= 90.00  beta= 90.00  gamma= 90.00

total energy= -2500.0000
```

- ▶ 配布ファイル中の file00.kkr を読み込んで内容を表示するスクリプト (test1.py) を作ってください

## 練習 (2)

- ▶ ファイル file00.kkr を読み込んで、文字列 total energy= が含まれている行だけを表示してください

解答例 (test2.py)

```
#!/usr/bin/env python
f = open("file00.kkr", "r")
for line in f:
    if "total energy=" in line:
        print line
f.close()
```

- ▶ " 検索文字列 " in 全文
  - ▶ in 演算子 特定の文字列が含まれている場合のみ True を返す

## 練習 (3)

- ▶ ファイル file00.kkr を読み込んで、全エネルギー (total energy) を変数 energy に保存して、出力するスクリプトを作ってください

解答例 (test3.py)

```
#!/usr/bin/env python
f = open("file00.kkr", "r")
for line in f:
    if "total energy=" in line:
        data = line.split("=")
        energy = float(data[1])
f.close()
print energy
```

- ▶ "文字列".split("記号")
  - ▶ 特定の記号が含まれている位置で文字列を分割する
  - ▶ total energy= -2500.0000 を=で分割すると ["total energy", "-2500.000"] のリストが変数 data に代入

## 練習 (4)

- ▶ ファイル file00.kkr を読み込んで、全エネルギー (total energy) と格子定数を抽出するスクリプトを作ってください

解答例 (test4.py)

```
#!/usr/bin/env python
f = open("file00.kkr", "r")
for line in f:
    if "total energy=" in line:
        data = line.split("=")
        energy = float(data[1])
    if "bravais=bcc" in line:
        data = line.split()
        a = float(data[2])
f.close()
print "%f %f" % (a, energy)
```

## 練習 (5)

ディレクトリ内に複数の.kkr ファイルがあるとき、すべての.kkr ファイルの全エネルギーと格子定数を表示するスクリプトを作ってください

—— 解答例 (test5.py) ——

```
#!/usr/bin/env python
import glob
ls = glob.glob("*.kkr")
for name in ls:
    f = open(name, "r")
    for line in f:
        if "total energy=" in line:
            data = line.split("=")
            energy = float(data[1])
        if "bravais=bcc" in line:
            data = line.split()
            a = float(data[2])
    f.close()
    print "%f %f" % (a, energy)
```

## 演習 1

### リスト

- ▶ 25.2, 16.8, 31.4, 23.9, 28.0, 22.5, 16.9 という値を持つ摂氏の温度リストを作り、*temps* 変数に代入してください
- ▶ あるリストメソッドを使って *temps* を昇順にソートしてください
- ▶ スライシングを使って、それぞれ 20 度未満、20 度以上の気温を集めた *cool\_temps* と *warm\_temps* を作ってください

教科書 p93 より

### 出力ファイル処理

- ▶ sample2/file00.out の total energy と格子定数を表示するスクリプトを考えてください
- ▶ sample2/\*.out のすべての total energy とその時の格子定数を表示するスクリプトを考えてください
- ▶ sample2/\*.out のなかで最小の total energy とその時の格子定数を求めるスクリプトを考えてください

## まとめ

- ▶ 今回はリストのスライス、メソッドについて説明しました
- ▶ 第一原理計算などの出力ファイルの処理に関する練習を行いました
- ▶ 次回は
  - ▶ 6 章 (条件分岐)
  - ▶ 7 章 (反復)
- ▶ を予定しています

## 演習 2

### xyz 形式へのコンバート

sample3.out に含まれている原子位置データを読み込み、xyz 形式で書き出すスクリプトを作ってください

—— 例 (sample3.out) ——

lattice parameter (alat) = 10.3935 a.u.

(中略)

2 H tau( 2) = ( 0.1154700 0.1154700 0.1154700 )

座標成分は格子定数 (今回は 10.3935 Bohr) を単位としており、また原子単位を Å に換算するためには 0.529 を乗する必要があります。このため、out ファイルの原子位置座標を Å 単位に換算するためには、

$$x = 0.1154700 \times 10.3935 \times 0.529 \quad (\text{\AA}) \quad (1)$$