An Introduction to Python Part 2

M. Uemoto

Yoshida Lab

July 9, 2013

本日の予定

- 基本文法（教科書第 3 章）
  - 変数型
  - 文字列型
- 補足
  - ファイル書き込み
  - VESTA について
- 演習問題

前回のおさらい

for ループ

```
for i in xrange(0, 10):
    print i
```

条件分岐

```
if x == 0:
    print "x is zero"
else:
    print "x is not zero"
```

関数定義

```
def f(x):
    return x*x
```

- インデントブロック（空白の深さを揃える）
- コロン「:」をつけ忘れない

前回の宿題

Even Fibonacci numbers

- Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:
- 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
- By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

- while 条件式:
  条件式が真 (True) である限り、直下のインデントブロックの処理を繰り返す。（ループ命令）

while ループの使用例

```
i = 0
while i < 10:
    print i
    i += 1
```

再帰による解答例（舩島さんより）

```
#!/usr/bin/env python

def fibonacci(n):
    if 1<= n <= 2 :
        return n
    else:
        return fibonacci(n-2) + fibonacci(n-1)

Threshold = 4000000
n = 1
m = fibonacci(2*n)
sumN = m
n += 1
while m <= Threshold:
    m = fibonacci(2*n)
    sumN += m
    n += 1

print sumN
```

変数とデータ型 (1)

- 変数
  - データを格納
- データには「型」が存在する
  - 数値
    - 整数 (int)
    - 浮動小数 (float)
    - 複素数 (complex)
  - 文字列 (str)
  - コンテナ（tuple, list, dict, set...)
  - クラス・オブジェクト
- 型によって使用出来る操作がきまる
  - 例）数値型：四則演算子 (+ - * /)、絶対値 (abs)...
  - 例）文字列：連結演算子 (+)、文字列長 (len)...

変数とデータ型 (2)

- type 関数
  変数のデータ型を表示する

整数型 (int)

```
>>> x = 1
>>> type(x)
<type 'int'>
```

浮動小数 (float)

```
>>> x = 1.0
>>> type(x)
<type 'float'>
```

複素数 (complex)

```
>>> x = 1.0+1.0j
>>> type(x)
<type 'complex'>
```

文字列型 (str)

文字列型の代入

```
>>> x = "Hello"
>>> print x
Hello
>>> y = 'World'
>>> print y
World
```

- 文字列 (str)··· 文字の配列
- シングルクォート「'」かダブルクオート「"」で囲む
- print 文で内容を表示する
- 文中にクオート記号が入る場合
  - シングルクォートとダブルクォートを使い分ける
    'He said "Hello!"'
  - エスケープ文字を使う（後ほど説明）

エスケープ文字

Python はバックスラッシュ(環境によっては円記号) で始まる文字を特殊な記号として処理する。

- 例) 「\'」はシングルクォート「'」として表示される

エスケープシーケンス

```
>>> print "123\"456\"789\n123\t456\\789"
123"456"789
123 456\789
```

- 「\"」,「\'」 クォテーション
- 「\\」 バックスラッシュ（または円記号）
- 「\n」 改行コード
- 「\t」 タブコード

数値型から文字列型への変換

フォーマット演算子

```
>>> "Answer is %d" % 1
'Answer is 1'
>>> "Answer is %f" % 1.0
'Answer is 1.000000'
>>> "Answer is %e" % 1.0
'Answer is 1.000000e+00'
```

- 文字列中の「%d」が数値に置き換えられる
- %d(整数)、%f(小数)、%e(指数表示)
- "Answer is %s" % "abc"
  %s は%演算子以降の文字列と置き換えられる
- その他の方法、str 関数

ファイル書き込み

- open("ファイル名", "w")
  - ファイルを書き込みモードで開く
- write("テキスト")
  - ファイルにテキストを書き足す
- close()
  - ファイルを閉じる

サンプル (test.txt へのデータ書き込み)

```
>>> f = open("test.txt", "w")
>>> f.write("Hello\n")
>>> f.write("World\n")
>>> f.close()
```

- 「f」ファイルハンドル、test.txt へのアクセス権限を持ったオブジェクト

結果 (test.txt)

```
Hello
World
```

入力ファイルの自動生成 (1)

create.py

```
#!/usr/bin/env python

for i in xrange(0, 10):
    a = 5.0 + i*0.01
    f = open("input%d.txt", "w")
    f.write("Unit vector\n")
    f.write("%f 0.0 0.0\n" % a)
    f.write("0.0 %f 0.0\n" % a)
    f.write("0.0 0.0 %f\n" % a)
    f.close()
```

- 格子定数 a を $5.0 \leq a < 5.1$ まで 0.01 きざみで増やす
- 生成したインプットファイルを input0.txt から input9.txt に保存する

入力ファイルの自動生成 (2)

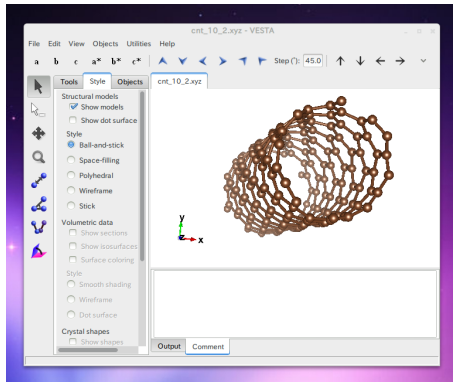
- for i in xrange(0, 10):
  ループ変数 $0 \leq i < 10$ の範囲で繰り返し
- f = open("input%d.txt" % i, "w")
  ファイルを開く、保存時のファイル名は「input 数字.txt」にする
- f.write("%f 0.0 0.0\n" % a)
  ファイルへ書き込みを行う「数値 0.0 0.0」という形で出力

スクリプトが自動生成するインプットファイルの一例

input1.txt

```
Unit vector
5.010000 0.0 0.0
0.0 5.010000 0.0
0.0 0.0 5.010000
```

VESTA



- 結晶構造の三次元データの可視化プログラム
- オープンソースで配布
- http://jp-minerals.org/vesta/jp/ (日本語版配布元)

VESTA/結晶構造のインプットファイル

NaCl 型結晶の原子位置データ (nacl.xyz)

nacl.xyz

```
64
NaCl Structure
Na 0.000000 0.000000 0.000000
Cl 0.000000 0.000000 5.630000
Na 0.000000 0.000000 11.260000
Cl 0.000000 0.000000 16.890000

(以下ずっと続く)
```

- 一行目、スーパーセル中の全原子数
- 二行目、タイトル
- 三行目、原子位置「元素 X 座標 Y 座標 Z 座標」、以下続く

nacl.py

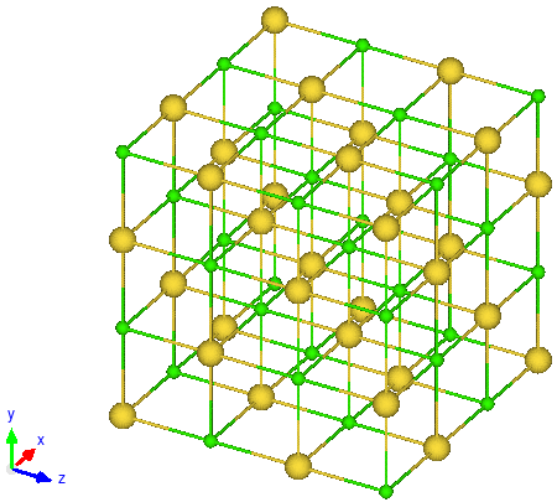
```
#!/usr/bin/env python

f = open("nacl.xyz", "w")
n = 4*4*4
f.write("%d\n" % n)
f.write("NaCl Structure\n")

for i in xrange(4):
    for j in xrange(4):
        for k in xrange(4):
            x = 5.63*i
            y = 5.63*j
            z = 5.63*k
            if (i+j+k)%2 == 0:
                f.write("Na %f %f %f\n" % (x, y, z))
            else:
                f.write("Cl %f %f %f\n" % (x, y, z))
f.close()
```

結果

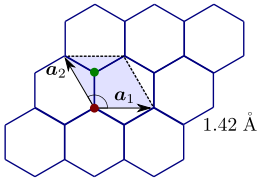
出来上がった nacl.xyz ファイルを VESTA で開く

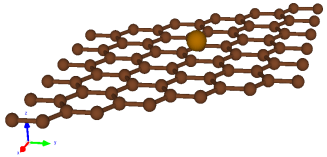


課題 1

グラフェン

- (1) グラフェンのスーパーセルにおけるの炭素原子位置を計算し、「graphen.xyz」のファイル名で保存するスクリプトを作成してください
- (2) さらに、六員環中心位置 (H-site) の上 z Å の位置に、鉄 (Fe) 原子が吸着されている状態を考えます。$1.0 \leq z < 1.5$ の範囲で、Fe 原子の表面からの距離を変えながら、さまざまな z の値における xyz ファイルを自動生成するスクリプトを考えてください。

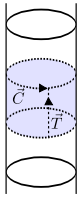


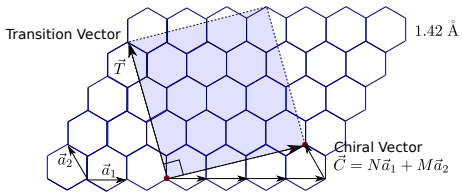


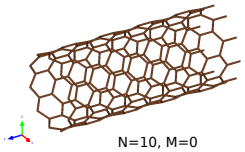
課題 2

カーボンナノチューブ

さまざまなカイラル指数 (N, M) の単層カーボンナノチューブの炭素原子位置を、「cnt_N_M.xyz」のファイル名で自動生成をおこなうスクリプトを考えてください。







- カイラルベクトル
  円筒軸に垂直に円筒面を 1 周するベクトル
  グラフェンの基本並進ベクトル $\vec{a}_1, \vec{a}_2$ を使って $\vec{C} = N\vec{a}_1 + M\vec{a}_2$
- (N, M) は整数（カイラル指数）