

開発演習 II(新納研) (Python の学習 2)

新納浩幸

1 本日の目標

- リスト
- 辞書

2 リスト

汎用的なデータ構造、すべてのものを扱える。

```
>>> a = [ 5, 12, 8 ]      # 数値のリスト
>>> b = [ 'a', 'b', 'c' ] # 文字のリスト
>>> c = [ "Shinnou", "Bai", "Zhao" ] # 文字列のリスト

>>> a[0], b[1], c[2] # 要素の参照は index 番号
(5, 'b', 'Zhao')
>>>

>>> d = [ a, b, c ]     # リストのリスト
>>> d
[[5, 12, 8], ['a', 'b', 'c'], ['Shinnou', 'Bai', 'Zhao']]

>>> d[1][2] # リストのリストの要素の参照
'c'

>>> e = a + b # リストの連結
>>> e
[5, 12, 8, 'a', 'b', 'c']

>>> len(e) # リストの要素数
6

>>> 10 in e # ある要素が存在するかの判定
False
>>> 'b' in e
True

# 要素の削除は del, pop, remove

>>> del e[2]
>>> e
[5, 12, 'a', 'b', 'c']
```

スライスの操作は重要！

```
>>> r1 = range(10)
>>> r1
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> r1[2:6] # index が 2 から (6-1) まで
[2, 3, 4, 5]

>>> r1[-1] # 要素の最後の index は -1
9

>>> r1[-2] # 要素の最後から 2 番目の index は -2
8

>>> r1[:] # a:b a から b
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> r1[:5] # :b 最初 から b
[0, 1, 2, 3, 4]
>>> r1[3:] # a: a から 最後
[3, 4, 5, 6, 7, 8, 9]
```

オブジェクトの型判定

```
>>> r1
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> isinstance(r1,list) # リストかどうか
True
>>> isinstance(r1[0],list)
False
>>> isinstance(r1[0],int) # 整数かどうか
True
```

リストを逆順に表示するプログラム

```
lst = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
for i in range(len(lst)):
    k = - i - 1
    print lst[k],

print
```

[実行結果例]

```
> python b1.py
g f e d c b a
```

課題9 入れ子になっているリスト中の全ての要素数がいくつになのかを調べるプログラムをかけ。

例えば

```
a = [[1, [2, [3, 4] [5, 6, [7, 8], 9, [10]]]]]]
```

の場合、要素数は 10 である。

課題10 前回利用した seiseki.dat の中身は以下。

```
> cat seiseki.dat
hanako 86 79 67 68
tarou 71 75 78 84
jirou 42 43 39 44
yasuko 62 58 98 95
hiroki 96 97 61 63
hiroko 39 33 45 50
kanako 50 53 64 72
minoru 78 66 52 47
ken 51 44 76 72
yuuko 89 92 93 91
```

このファイルを読み込んで、各人毎に

[名前, 第1科目の点, 第2科目の点, 第3科目の点, 第4科目の点, 4科目の合計点]

のリストを作成、このリストを要素としたリスト (10個のリストを要素とするリスト) を作成し、表示せよ。

sorted の追加の説明。key を使って、柔軟なソートが可能。

```

>>> lst = [ ['a', 10], ['b', 8], ['c', 7], ['d', 15] ]
>>> lst2 = sorted(lst, key=lambda x: x[1])  ## 要素の1番目でソート
>>> lst2
[['c', 7], ['b', 8], ['a', 10], ['d', 15]]

>>> lst3 = sorted(lst, key=lambda x: -x[1])  ## 降順はマイナス
>>> lst3
[['d', 15], ['a', 10], ['b', 8], ['c', 7]]

```

課題 11 課題 10 と先の説明を利用して、課題 8 をもう 1 回やってみよ。

3 辞書

連想配列 あるいは Hash と呼ばれるデータ構造。Python では「辞書」と呼ぶ。

リストの要素を参照するには、参照のための key としてリスト中の位置を表す index（整数値）が使われる。参照のための key として、文字列を使えるようにしたものが辞書、、と捉えていて問題ない。

```

dic = {}

dic['year'] = 2016
dic['month'] = 10
dic['day'] = 42
dic['hour'] = 14
dic['min'] = 20

for k in dic:      # 辞書の要素の列挙
    print k, dic[k]

```

[実行結果例]

```

> python b2.py
min 20
month 10
day 42
hour 14
year 2016

```

seiseki.dat を辞書に読み込んで、名前を key にして、その成績を表示するプログラム。

```
import sys

argsv = sys.argv
argc = len(argsv)

dic = {}

f = open(argsv[1], 'r')
line = f.readline()
while line:
    line = line.rstrip() ## <-- chomp
    lst = line.split()
    dic[lst[0]] = line    ## key は名前、value は1行全部
    line = f.readline()
f.close

while True:
    key = raw_input("Please input a name >> ")
    if key in dic:
        print dic[key]
    elif key == "":     ## 改行のみで終了
        break
    else:
        print "no data"
```

[実行結果例]

```
> python b3.py seiseki.dat
Please input a name >> hanako
hanako 86      79      67      68
Please input a name >> taro
no data
Please input a name >>     ## 改行のみで終了

>
```

eng.txt を読み込んで、利用されている単語を列挙するプログラム。単語は空白か改行で区切られたものと定義する。細かいことは気にしない。

```
import sys

argsv = sys.argv
argc = len(argsv)

dic = {}

f = open(argsv[1], 'r')
line = f.readline()
while line:
    line = line.rstrip()
    for word in line.split():
        if word not in dic:
            dic[word] = 1
    line = f.readline()
f.close

id = 1
for word in dic:
    print id, word
    id += 1
```

[実行結果例]

```
> python b4.py eng.txt
1 all
2 reform,
3 trainwreck,
4 government.
5 Sloppy,

. . .

560 ideological
561 So
562 2008
563 restored
564 once

>
```

課題 12 eng.txt を読み込んで、利用されている単語とその頻度を求め、頻度の高い順に表示せよ。