

みんなのPython

-13- 標準ライブラリを使う

山木翔馬

collections.OrderedDict

- 順序を持ったディクショナリ

```
>>> from collections import OrderedDict
>>> od = OrderedDict() #インスタンスを作る
>>> od[ 'a' ] = 'A'
>>> od[ 'b' ] = 'B'
>>> od[ 'c' ] = 'C'
>>> od
OrderedDict( [ ( 'a', 'A' ), ( 'b', 'B' ), ( 'c', 'C' ) ] )
```

Collections.defaultdict

- キーに対応する値のデフォルト値を登録できる

```
>>> animals = [ ('猫', '三毛'), ('犬', 'コーギー'),  
                ('猫', 'シャム'), ('犬', 'ダックス') ]  
  
>>> d = {}  
>>> for k, v in animals :  
...     d.setdefault( k, [] ).append( v )  
  
...  
>>> d  
{ '猫' : [ '三毛', 'シャム' ],  
  '犬' : [ 'コーギー', 'ダックス' ] }
```

bisect

- ソート済リストの要素を常にソートした状態に保つ

```
>>> import bisect
```

```
>>> a = sorted( [ 2, 1, 6, 5 ] ) # [ 1, 2, 5, 6 ]
```

```
>>> bisect.insort( a, 3 ) # [ 1, 2, 3, 5, 6 ]
```

```
>>> bisect.bisect( a, 4 )
```

```
3
```

日時データを扱うモジュール

time モジュール

- 文字列から日時に関するデータを得る
- 日時を表す文字列を得る
- 一定時間処理を停止する sleep関数など

datetime モジュール

- 1970年以前、2036年以降など広範囲を扱える
- 日付や時刻を使った演算や比較が行いやすい

基本的に datetimeモジュールを使う

datetime.dateクラス

```
>>> from datetime import date
```

```
>>> today = date.today( )
```

```
>>> today.year
```

```
2015
```

```
>>> today.month
```

```
6
```

```
>>> today.day
```

```
19
```

datetime.datetimeクラス

```
>>> from datetime import datetime
```

```
>>> t = datetime( 12, 34, 56 )
```

```
>>> t.hour
```

```
12
```

```
>>> t.minute
```

```
34
```

```
>>> t.second
```

```
56
```

datetime.datetimeクラス

```
>>> from datetime import datetime
>>> now = datetime.now()
>>> now
datetime.datetime(2015, 6, 19, 5, 6, 0, 11138)
```

- アトリビュートに year, month, day, hour, minute, second, microsecond を持つ

datetime.timedeltaクラス

```
>>> from datetime import timedelta
```

```
>>> print timedelta( second=1000000 )
```

```
11 days, 13:46:40
```

```
>>> timedelta(days=8, hours=16, minutes=30)
```

```
- timedelta(days=6, hours=20, minutes=40)
```

```
datetime.timedelta(1, 71400)
```

OSモジュール

- OS依存の処理を行う
- `os.chdir`, `os.getcwd`, `os.remove`, `os.mkdir...`などのファイル操作
- `os.walk(path [, topdown=True])`
 - より深い階層へ (or より浅い階層へ) 向かってディレクトリを移動する
 - 処理中のディレクトリパス、処理中のディレクトリに含まれるディレクトリ名一覧、およびファイル名一覧を取得
- `os.system(command)`
 - 子プロセスとして引数のコマンドを実行

mathモジュール

- 数学関数を扱うモジュール
- `math.pi`, `math.e...` などの定数
- `math.pow`, `math.sqrt`, `math.sin...` などの関数

randomモジュール

- ランダムな値を用いるための関数など
- randint(a, b)
 - a以上b以下のランダムな整数を返す
- random()
 - 0以上1以下のランダムな浮動小数点数を返す
- shuffle(シーケンス [, random])
 - シーケンスの要素をランダムに入れ替える。引数として渡したシーケンス自体を変更する。

urllibモジュール

- インターネット上のデータを取得する
- `request.urlopen(url [, post_data [, timeout]])`
 - WebやFTPなどからデータを取得して、データを読み込み専用のファイル風オブジェクトに格納する
 - ファイル風オブジェクト(R)はR.read, R.readlineなどファイルオブジェクトと同じ読み込み関数を持つ

```
>>> from urllib import request
```

```
>>> data = request.urlopen( url ).read( )
```

io.StringIOクラス

- 文字列のバッファを使ってファイル風オブジェクトを作る
- ファイルの読み書きをメモリ上でシミュレーション

```
>>> from io import StringIO
>>> f = StringIO( )
>>> f.write( "a" * 10 )
>>> f.seek( 0 )
>>> f.read( )
'aaaaaaaaaaaa'
```

csvモジュール (csv.reader)

- `csv.reader(ファイルオブジェクト[, dialect='excel'[, fmtparam]])`
 - CSVファイルをファイルオブジェクトとして呼び出す
 - 返り値はreaderオブジェクト (イテレータの一種)

```
import csv
csvfile = open( "test.csv", encoding="utf-8")
for row in csv.reader( csvfile )
    print row
```

csvモジュール (csv.writer)

- `csv.writer(ファイルオブジェクト [, dialect='excel' [, fmtparam]])`
 - csvファイルを書き出す
- `csv.writerow(シーケンス)`
 - 引数に与えたシーケンスをファイルオブジェクトに書き出す
- `csv.writerows(シーケンス)`
 - シーケンスを要素に含むシーケンスを引数に渡すと、要素を区切って複数の行を書き出す

shelveモジュール

- デクショナリをファイルに書き出す

```
import shelve
d = shelve.open( "shelvetest" )
d.update( { "one" : 1, "two" : 2 } )
d.close()
```

```
d2 = shelve.open( "shelvetest" )
d2
{ 'one' : 1, 'two' : 2 }
```

pickleモジュール

- オブジェクトの永続化、シリアライズ
- shelveよりも多くの種類のオブジェクトを扱える

```
import pickle
```

```
o = [ 1, 2, 3, { "one" : 1, "two" : 2 } ]
```

```
pickle.dump( o, open( "pickle.dump", "wb" ) )
```

```
o2 = pickle.load( open( "pickle.dump", "rb" ) )
```

```
o2
```

```
[ 1, 2, 3, { 'one' : 1, 'two' : 2 } ]
```

jsonモジュール

- JSONデータを扱うモジュール
- `json.loads(s)`
 - JSON文字列をPythonのデータ型に変換
 - `json.load(fp)` でJSONファイルも扱える
- `json.dumps(Pythonオブジェクト [, オプション...])`
 - PythonオブジェクトをJSON文字列に変換
 - `json.dump(Pythonオブジェクト, fp)` でJSON文字列をファイルに書き出す

pprintモジュール

- リストやディクショナリなどのデータを綺麗に整えて出力する

```
from pprint import pprint
l = [ list( range( 10 ) ) * 3 ]
pprint( l )
[[ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
 [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
 [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]]
```

課題13

- collectionsモジュールのCounterクラスについて調べ、英文中の単語を数えるプログラムを作成せよ

```
text = "dog cat dog dog dog cat"
```

↓

```
Counter({'dog': 4, 'cat': 2})
```