Python サンプルコード

[Python 2.5][言語まとめ Python][Python 標準ライブラリ概観]

ライブラリ

・Python 標準ライブラリ概観

ライブラリ・フレームワーク	内容
Django	Web アプリケーションフレームワーク
Google App Engine	Google Web アプリケーションフレームワーク
Python Imaging Library	イメージング
Beautiful Soup	HTML 解析
Python Win32 Extensions	COM 操作
pyExcelerator	Excel 操作

はじめに

HELP

・help 関数の使い方

ビルトインを調べる

```
>>> help(__builtins__)
>>> dir(__builtins__)
```

属性の確認

・" オブジェクト " に調べたいオブジェクトを設定

```
>>> dir( オブジェクト )
```

ヘッダー

```
#!python2.5
# -*- coding: utf-8 -*-
```

オブジェクト

オブジェクトの型

API	概要
type()	【組み込み関数】オブジェクトの型を返す

```
import os
print 'os type:%s' %(type(os))
fils = os.listdir('c:\work')
print 'os.listdir type:%s' %(type(fils))
print 'listdir[0] type:%s' %(type(fils[0]))
```

結果

```
os type:<type 'module'>
os.listdir type:<type 'list'>
listdir[0] type:<type 'str'>
```

ディレクトリとファイル

ディレクトリを再帰しながらファイルをリスト

```
def check_dir(dir):
    for o in glob.glob(os.path.join(dir,"*")):
        if os.path.isdir(o):
            check_package(o)
        else:
           print o
```

ディレクトリの内容を表示

API	概要
os.listdir	ディレクトリ内容のリストを取得

```
import os
fs = os.listdir('c:\work')
for f in fs:
  print f
```

ディレクトリの内容を表示 (2)

API	概要
os.path.isdir	パスのディレクトリ判定
os.path.exists	パスの存在判定
print 'C 言語スタイルの書式文字 ' %(パラメータ)	書式付出力

```
import os
basedir = 'c:\work'
fils = os.listdir(basedir)
print 'dir\texists\tpath'
print '------------------'
for fi in fils :
  p = basedir + os.sep + fi
  print '%s\t%s\t%s' %(str(os.path.isdir(p)), str(os.path.exists(p)), p)
```

Python 文字コードを指定してファイルを開く

・Python 文字コードを指定してファイルを開く

ファイルの内容を表示

・ディレクトリに含まれるファイルの内容を出力

API	概要
os.chdir	カレントディレクトリの変更
os.path.abspath	絶対パスの取得
open	【組み込み関数】ファイルを開く

print の末尾に "," で、改行出力を抑制

```
import os
os.chdir('c:\work')
fils = os.listdir('.')
for fi in fils:
  p = os.path.abspath(fi)
  if not os.path.isdir(p) :
    print '===== %s =====' %(fi)
    f = open(p)
    try:
      for line in f:
        print line,
    finally:
      f.close()
```

ファイルの情報を取得

API	概要
os.stat	与えられたパスに stat システムコールを実行
time.ctime	エポックタイムからの秒数をローカル時間に変換

```
>>> import time
>>> import os
>>> r = os.stat(r'/test.txt')
>>> r
nt.stat_result(st_mode=33206, st_ino=0L, st_dev=0, st_nlink=0, st_uid=0, st_gid=0, st_size=4508L,
st_atime=1266996480L, st_mtime=1266452528L, st_ctime=1251076742L)
>>> ct = time.ctime(r.st_ctime)
>>> ct
'Mon Aug 24 10:19:02 2009'
```

絶対パスからファイル名のみを取得

・os.path.basename

```
>>> import os
>>> os.path.basename('c/work/test.txt')
'test.txt'
```

パスを生成

API	概要
os.path.join	2 つ以上のパスを結合する。必要なら \ を挿入

```
join(a, *p)
```

```
>>> import os
>>> p = os.path.join('c:\\', 'work', 'test')
>>> print p
```

```
c:\work\test
```

ディレクトリ判定

API	概要
os.path.isdir(path)	ディレクトリか否かを判定

```
>>> import os
>>> os.path.isdir('c:\work')
True
```

ファイルをマージ

import os

```
import codecs

def csv_file_mearge(baseDir):
    encode = "shift_jis"
    subDir = "out"
    repName = "mearged.txt"
    os.makedirs(os.path.join(baseDir,subDir),exist_ok=True)

    with codecs.open(os.path.join(baseDir,subDir,repName), 'w', encode) as outf:
        files = os.listdir(baseDir)
        for file in files:
            print(file)
            abspath = os.path.join(baseDir,file)
            if os.path.isfile(abspath):
                with codecs.open(abspath, 'r', encode) as inf:
                    for line in inf:
                        outf.write(line)

if __name__ == "__main__":
    csv_file_mearge("c:\work\")
```

ディレクトリの走査 (再帰)

例 1

```
import os
import pprint

def trav(path, fn=''):
  l = []
  l.append(fn)
  d = os.path.join(path,fn)
  if os.path.isdir(d):
    for f in os.listdir(d):
      l.append(trav(d,f))
  return l

l = trav('C:\\', 'Python25')
pprint.pprint(l)
```

例 2

```
# -*- coding: utf-8 -*-
import os

class FileTrav(object):
    def traverse(self, working_dir, depth=0):
        path = os.path.abspath(working_dir)
        for f in os.listdir(path):
            fl = os.path.join(path, f)
            print '%s%s' % ('\t' * depth, fl)
```

```
            if os.path.isdir(fl):
                self.traverse(working_dir=fl, depth=depth + 1)

if __name__ == '__main__':
    tp = FileTrav()
    tp.traverse(working_dir=os.getcwd())
```

・Python ファイルの文字コード

オブジェクト指向

・Python サンプルコード オブジェクト指向

コレクション

リスト

■結合

```
>>> l = []
>>> l.append('a')
>>> l.append('b')
>>> l
['a', 'b']
>>> ','.join(l)
'a,b'
```

リスト同士を結合する場合、extend を利用する

■スライス

API	概要
list()	リストの生成、[リストの内容, ・・・] で宣言と同時に初期化
[開始位置 : 終了位置 :STEP]	スライス。開始位置から終了位置まで STEP ごとに切り出す

```
#l = list()
l = [6,5,2,3,1,4]
l.append(7)
l.append(8)
l.sort()
print l
print l[::2]
print l[1::2]
```

結果

```
[1, 2, 3, 4, 5, 6, 7, 8]
[1, 3, 5, 7]
[2, 4, 6, 8]
```

コンテナオブジェクトからリストを作成

・文字列は文字のコンテナオブジェクト

```
msg = 'this is message.'
l = list(msg)
print l
```

結果

```
['t', 'h', 'i', 's', ' ', 'i', 's', ' ', 'm', 'e', 's', 's', 'a', 'g', 'e', '.']
```

タプル

・リストや配列と異なり、用途や型が異なるオブジェクトをひとつにまとめるために使われる
・C 言語の構造体を匿名にしたようなものと考えることができる
・ひとつの「かたまり」として変数に代入できる
・関数の返り値として使うこともできる。これによって、複数の値を返す関数を実現することができる
・リストと同様に、個々の要素を添え字によって参照できる
・あくまで「ひとつの値」であるため、一度構築したら中の値を変更することができない

```
t1 = 1,'one',
t2 = 2,'two',
t3 = 3,'tree',

t  = t1,t2,t3,
print t
```

結果

```
((1, 'one'), (2, 'two'), (3, 'tree'))
```

Set

■作成

```
s = set('aaaabbbbccccddd')
print s
```

■操作

追加

```
s.add('123')
```

共通集合

```
s & set([1,2,3])
```

その他

・union: 和集合を返す
・intersection: 共通集合を返す

・difference: 差集合を返す
・symmetric_difference: 対称的差集合を返す
・add: 追加
・remove: 削除

結果

```
set(['a', 'c', 'b', 'd'])
```

辞書

■キーとそれに対応する値を同時に取り出す

・iteritems() メソッドを使う

```
>>> m = {'a':1,'b':2,'c':1,'d':2}
>>> for k, v in m.iteritems():
...     print '%s,%s' % (k, v)
...
a,1
c,1
b,2
d,2
```

■条件にあった要素をリスト内包表記で取り出す

```
>>> m = {'a':1,'b':2,'c':1,'d':2}
>>> m2 = dict([(k, v) for k, v in m.iteritems() if v == 1])
>>> m2
{'a': 1, 'c': 1}
```

■辞書に辞書を追加する

```
>>> m = {'a':1,'b':2}
>>> m1 = {'c':3,'d':4}
>>> m.update(m1)
>>> m
{'a': 1, 'c': 3, 'b': 2, 'd': 4}
```

データベース

データベース (SQLite) の使用

・http://www.python.jp/doc/release/lib/module-sqlite3.html

```
#!python2.5
# -*- coding: utf-8 -*-
import sqlite3

con = sqlite3.connect('/work/py/test.db')

#create database in RAM
#con = sqlite3.connect(':memory:')

con.execute("create table test (id  text, value text, note text)")

con.execute("insert into test values('1','aaa','01')")
con.execute("insert into test values('2','bbb','02')")
con.execute("insert into test values('3','ccc','01')")

con.commit()
```

```
p = ('01',)
c = con.cursor()
c.execute("select * from test where note=?", p)

for r in c:
  print r

con.close()
```

結果

```
(u'1', u'aaa', u'01')
(u'3', u'ccc', u'01')
```