

Kurs języka Python

Wykład 6.

Marcin Młotkowski

23 listopada 2009

- 1 Pliki tekstowe
- 2 Trwałość obiektów
- 3 CSV
- 4 Strumienie
- 5 Pliki *.ini
- 6 Pliki rekordów

Operacje na plikach

Otwarcie i zamknięcie pliku

```
fh = open('plik', 'r')
```

```
...
```

```
fh.close()
```

Atrybuty otwarcia

'r'	odczyt
'w'	zapis
'a'	dopisanie
'r+'	odczyt i zapis
'rb', 'wb', 'ab'	odczyt i zapis binarny (tylko Windows i Macintosh)

Metody czytania pliku

Odczyt całego pliku

```
fh.read()
```

Odczyt tylko size znaków

```
fh.read(size)
```

Odczyt wiersza, wraz ze znakiem '\n'

```
fh.readline()
```

Zwraca listę odczytanych wierszy

```
fh.readlines()
```

Odczyt pliku

Przykład

```
fh = open('test.py', 'r')
while True:
    wiersz = fh.readline()
    if len(wiersz) == 0: break
    print wiersz,
fh.close()
```

Inny przykład

```
fh = open('test.py', 'r')
for wiersz in fh:
    print wiersz,
```

Odczyt pliku

Przykład

```
fh = open('test.py', 'r')
while True:
    wiersz = fh.readline()
    if len(wiersz) == 0: break
    print wiersz,
fh.close()
```

Inny przykład

```
fh = open('test.py', 'r')
for wiersz in fh:
    print wiersz,
```

Zapis do pliku

```
fh.write('dane zapisywane do pliku\n')  
fh.writelines(['to\n', 'są\n', 'kolejne\n', 'wiersze\n'])
```

Zamykanie pliku

Uwaga

Zawsze należy zamykać pliki.

Przykład

try:

```
fh = open('nieistniejacy', 'r')  
data = fh.read()
```

finally:

```
fh.close()
```

Alternatywne zamykanie pliku

```
del fh
```

Zamykanie pliku

Uwaga

Zawsze należy zamykać pliki.

Przykład

try:

```
fh = open('nieistniejacy', 'r')  
data = fh.read()
```

finally:

```
fh.close()
```

Alternatywne zamykanie pliku

```
del fh
```

Zamykanie pliku

Porada

```
with open('nieistniejący', 'r') as fh:  
    data = fh.read()
```

Uwaga

Dopiero od 2.5, przy włączonej opcji `with_statement`.
Dostępne na stałe od 2.6.

Zamykanie pliku

Porada

```
with open('nieistniejący', 'r') as fh:  
    data = fh.read()
```

Uwaga

Dopiero od 2.5, przy włączonej opcji `with_statement`.
Dostępne na stałe od 2.6.

Formaty danych

- Pliki tekstowe
- Pliki z rekordami
- Pliki CVS
- Pliki *.ini
- XML
- ...

Przechowywanie obiektów w pliku: pakiet pickle

Zapis obiektu

```
import pickle
obj = Obj()

fh = open('plik.obj', 'w')
pickle.dump(obj, fh)
fh.close()
```

Odczyt obiektu

```
fh = open('plik.obj', 'r')
obj = pickle.load(fh)
fh.close()
```

Comma Separated Values — CSV

Dane osobowe

```
"imie1", "nazwisko1", 2001-01-01, 3  
"imie2", "nazwisko3", 2009-11-23, 2
```

Dane giełdowe

```
04PRO,2009-11-10 13:17:33.0,C/P/M,2.97,2.97,3.0,2.92,2.93  
05VICT,2009-11-10 13:18:01.0,C/P/S,0.84,0.84,0.86,0.84,0.85  
06MAGNA,2009-11-10 13:18:43.0,C/P/S,1.15,1.15,1.17,1.14,1.14
```

Parametry formatu

delimiter	separator, np. ',' ';' ':'
lineterminator	koniec wiersza
quotechar	znak cudzysłowu
quoting	kiedy ujmować pola w cudzysłów; QUOTE_ALL, QUOTE_NONNUMERIC, QUOTE_NONE

CSV - dialekty

- Dialekt: domyślne parametry, np. dialekt excel
- Możliwość dodania własnego dialektu

Odczyt CSV

Przykład

```
import csv
reader = csv.reader(
    open('/etc/passwd', 'r'),
    delimiter=':',
    quoting=csv.QUOTE_NONE)
for row in reader:
    print row
```

Zapis w formacie CSV

Przykład

```
import csv
data = [ [1, 'Kubus'], [2, 'Puchatek'] ]
writer = csv.writer(
    open('out', "w"),
    dialect=csv.excel)
writer.writerows(data)
```

CSV — słowniki

```
Data;FS;FO;FOE;FSW;FZ;FPA;FRP;FANE;  
2009-05-05;1416.17;183.01;101.80;123.63;  
2009-05-04;1416.03;183.11;101.95;123.48;  
2009-05-02;1415.79;183.06;101.93;123.43;
```

Przetwarzanie słowników

```
fh = open('notowania.csv', "r")
reader = csv.DictReader(fh, delimiter=';')
for row in reader:
    for k in row.keys():
        print k, row[k]
```

CSV, słowniki cd.

Własne nazwy kolumn

```
fh = open('notowania.csv', "r")
klucze = ['lp', 'name']
reader = csv.DictReader(fh, fieldnames= klucze)
for row in reader:
    print row['lp'], row['name']
```

Strumienie; motywacje

```
def przegladaj_notowania(zrodlo):  
    reader = csv.DictReader(zrodlo, delimiter=';')  
    for row in reader:  
        for k in row.keys():  
            print k, row[k]
```

Wywołanie funkcji

```
przegladaj_notowania('dane.csv')  
  
url = 'http://www.notowania.pl/dane.csv'  
przegladaj_notowania(url)
```

Rodzaje strumieni danych

- Plik dyskowy
- Sieć komputerowa (url)
- String
- ...

Własności strumieni

- implementacja kolekcji
- `.read()`
- `.read(size)`
- `.write(buf)`
- `.close()`
- ...

Implementacja funkcji

```
def przegladaj_notowania(zrodlo):  
    reader = csv.DictReader(zrodlo, delimiter=';')  
    for row in reader:  
        for k in row.keys():  
            print k, row[k]
```

Plik dyskowy

```
in_stream = open('dane.csv', 'r')
przeogladaj_notowania(in_stream)
in_stream.close()
```

Internet

```
import urllib
url = 'http://www.notowania.pl/dane.csv'
in_stream = urllib.urlopen(url)
przeogladaj_notowania(in_stream)
in_stream.close()
```

String jako strumień

String jako strumień

```
str = '2006-11-13;12.45;45.78'  
import StringIO  
in_stream = StringIO.StringIO(str)  
przegladaj_notowania(in_stream)  
in_stream.close()
```

Pliki *.ini

Przykład

```
[window]
height=487
width=601
state=0

[time_plugin]
prompt_type=1
```

Biblioteka ConfigParser

Odczyt wartości

```
import ConfigParser
ini = ConfigParser.ConfigParser()
print ini.sections()
print ini.getint('window', 'height')
print ini.items('window')
```

Modyfikacja wartości

```
ini.set('window', 'height', 100)
```

Usuwanie

```
ini.remove_option('window', 'height')
ini.remove_section('window')
```

Biblioteka ConfigParser

Odczyt wartości

```
import ConfigParser
ini = ConfigParser.ConfigParser()
print ini.sections()
print ini.getint('window', 'height')
print ini.items('window')
```

Modyfikacja wartości

```
ini.set('window', 'height', 100)
```

Usuwanie

```
ini.remove_option('window', 'height')
ini.remove_section('window')
```

Biblioteka ConfigParser

Odczyt wartości

```
import ConfigParser
ini = ConfigParser.ConfigParser()
print ini.sections()
print ini.getint('window', 'height')
print ini.items('window')
```

Modyfikacja wartości

```
ini.set('window', 'height', 100)
```

Usuwanie

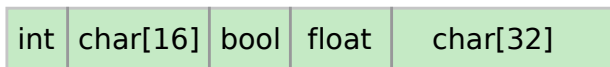
```
ini.remove_option('window', 'height')
ini.remove_section('window')
```

Zapis konfiguracji

```
fh = open('konfig.ini', 'w')  
ini.write(fh)  
fh.close()
```

Format rekordu

PLIK



REKORD

Konwersja

z rekordu na string

```
import struct
numrec, lenheader = struct.unpack('<xxxxLH22x', f.read(32))
name, typ, size, deci = struct.unpack('<11sc4xBB14x', rec)
```

na format rekord

```
hdr = struct.pack('<BBBBLHH20x',
                  ver, yr, mon, day, numrec, lenheader, lenrecord)
```

Znaki formatujące

'x'	nieistotny bajt, np. '4x' == 'xxxx'
'c'	znak
'i'	int
s', 'p'	stringi
<liczba>	powtórzenie wzorca

Dodatkowe znaki

'<'	little endian
'>'	big endian