

Αρχεία

Μιχάλης Δρακόπουλος

Ιούνιος 2020

ΑΡΧΕΙΑ

Αποθήκευση δεδομένων για είσοδο/έξοδο τους σε προγράμματα Python.

Είναι ενσωματωμένος τύπος της γλώσσας. Η Python: - διαχειρίζεται τα αρχεία ως **αντικείμενα** - παρέχει συναρτήσεις και μεθόδους για επεξεργασία αρχείων

Είδη αρχείων

- *Αρχεία κειμένου* - Το περιεχόμενό τους είναι συμβολοσειρές τύπου `str`
- *Δυναμικά αρχεία* - Το περιεχόμενό τους είναι συμβολοσειρές του ειδικού τύπου `bytes`

Δημιουργία αντικειμένου για διαχείριση αρχείου - η συνάρτηση `open`

Στην Python, δημιουργούμε ένα αντικείμενο `file_object` για να διαχειριστούμε με αυτό ένα εξωτερικό αρχείο `filename`:

```
file_object = open(filename, mode)
```

- Η πρώτη παράμετρος `filename` είναι μια συμβολοσειρά με το όνομα του αρχείου στον δίσκο του υπολογιστή.
- Η δεύτερη παράμετρος `mode` είναι μια συμβολοσειρά που καθορίζει το είδος της επεξεργασίας.
 - `r` άνοιγμα υπάρχοντος αρχείου για είσοδο (`read`) - είναι η default τιμή
 - `w` δημιουργία νέου αρχείου για έξοδο (`write`)
 - `a` άνοιγμα αρχείου για προσθήκη νέων δεδομένων στο τέλος του (`append`)

```
[1]: f = open('temp.txt', 'w')
```

Επεξεργασία με αρχεία

Γίνεται με κλήση **μεθόδων**

```
file_object.method()
```

Η μέθοδος `close`

Την καλούμε όταν ολοκληρωθεί η επεξεργασία του αρχείου: - Τερματίζει τη σύνδεση του προγράμματος με το εξωτερικό αρχείο στον δίσκο - Αποδίδει το αρχείο στο λειτουργικό σύστημα - Μεταφέρει μη καταχωρημένες εγγραφές (λόγω `buffering`) από τη μνήμη στον δίσκο

```
[2]: f.close()
```

Εγγραφή σε αρχεία κειμένου

```
[3]: f = open('myfile.txt', 'w')
```

Γράφουμε πάντα **συμβολοσειρές** που τις έχουμε μορφοποιήσει εκ των προτέρων. Για εγγραφές σε αρχεία, η Python δεν μετατρέπει αυτόματα δεδομένα άλλων τύπων σε συμβολοσειρές (όπως συμβαίνει με την `print`). Συνεπώς πρέπει να έχουμε **μετατρέψει σε συμβολοσειρές** όλα τα αντικείμενα που θέλουμε να εγγραφούν στο αρχείο.

Η μέθοδος `write`

```
[4]: f.write(5)
```

```
-----  
  
TypeError                                Traceback (most recent call last)  
  
<ipython-input-4-a88829d9ee54> in <module>()  
----> 1 f.write(5)  
  
TypeError: write() argument must be str, not int
```

```
[5]: f.write(str(5))
```

```
[5]: 1
```

```
[6]: for i in range(1,6):  
      f.write('Line ' + str(i))
```

```
[7]: f.close()
```

Η εγγραφή είναι **σειριακή**: οι χαρακτήρες της συμβολοσειράς γράφονται ένας-ένας στο αρχείο. Δεν έχουμε ζητήσει να γραφτεί ο χαρακτήρας νέας γραμμής `\n`

```
[8]: f = open('myfile.txt', 'w')  
for i in range(1,4):  
    f.write('Line ' + str(i) + '\n')  
f.close()
```

Ανάγνωση από αρχείο

Διαβάζουμε **συμβολοσειρές** από τις οποίες πρέπει να εξάγουμε τα δεδομένα που μας ενδιαφέρουν (με μεθόδους **συμβολοσειρών**) και να τα μετατρέψουμε στους τύπους που μας ενδιαφέρουν (με **συναρτήσεις μετα-**

τροπής)

Η μέθοδος read: όλο το αρχείο

```
[9]: f = open('myfile.txt')
s = f.read()
f.close()
s
```

```
[9]: 'Line 1\nLine 2\nLine 3\n'
```

```
[10]: print(s)
```

```
Line 1
Line 2
Line 3
```

Η μέθοδος readline: γραμμή-προς-γραμμή

```
[11]: f = open('myfile.txt')
```

```
[12]: f.readline()
```

```
[12]: 'Line 1\n'
```

```
[13]: f.readline()
```

```
[13]: 'Line 2\n'
```

```
[14]: f.readline()
```

```
[14]: 'Line 3\n'
```

```
[15]: f.readline()
```

```
[15]: ''
```

```
[16]: f.close()
```

Η μέθοδος readlines: οι γραμμές στοιχεία μιας λίστας

```
[17]: f = open('myfile.txt')
L = f.readlines()
f.close()
```

```
[18]: L
```

```
[18]: ['Line 1\n', 'Line 2\n', 'Line 3\n']
```

Με επανάληψη Τα αρχεία τηρούν το **πρωτόκολλο επανάληψης**

```
[19]: f = open('myfile.txt')
      for ln in f:
          print(ln[:-1], int(ln[-2])**3)
      f.close()
```

Line 1 1
Line 2 8
Line 3 27

Αρχεία και εξαιρέσεις

Η ανάγνωση και η εγγραφή μπορεί για διάφορους λόγους να προκαλέσουν εξαιρέσεις. Για παράδειγμα:

- Το αρχείο για ανάγνωση μπορεί να μην υπάρχει
- Το αρχείο για εγγραφή να μην μπορεί να δημιουργηθεί
- κλπ

Μπορούμε να χειριστούμε συνολικά τις εξαιρέσεις αυτές με `try` και την εξαίρεση με όνομα `OSError` (`IOError` σε παλιότερες εκδόσεις της Python).

Αποθήκευση αντικειμένων σε δυαδικά αρχεία - Σειριοποίηση: `pickle`

```
[20]: D = {'name': 'Bob', 'age': 42}
      L = [21, 34, 82, 30, 92, 105]
```

```
[21]: F = open('datafile.pkl', 'wb')
```

```
[22]: import pickle
      pickle.dump(D, F)
      pickle.dump(L, F)
      F.close()
```

```
[23]: F = open('datafile.pkl', 'rb')
      DD = pickle.load(F)
```

```
[24]: DD
```

```
[24]: {'name': 'Bob', 'age': 42}
```

```
[25]: LL = pickle.load(F)
```

```
[26]: LL
```

```
[26]: [21, 34, 82, 30, 92, 105]
```

Παραδείγματα

Θα χρησιμοποιήσουμε το εξής αρχείο:

```
[27]: print(open('grail.txt').read())
```

```
We're Knights of the Round Table
We dance whene'er we're able
We do routines and chorus scenes
With footwork impeccable
We dine well here in Camelot
We eat ham and jam and Spam a lot.
-- Monty Python
```

Αριθμός γραμμών, λέξεων και χαρακτήρων σε ένα αρχείο

```
[28]: def wc(filename):
    try:
        f = open(filename, 'r')
    except OSError as err:
        print(err)
    else:
        lines, words, chars = 0, 0, 0
        for line in f:
            lines += 1
            words += len(line.split())
            chars += len(line)
        finally:
            f.close()
    return lines, words, chars
```

```
[29]: wc('grail.txt')
```

```
[29]: (7, 38, 216)
```

Κωδικοποίηση rot13

Κωδικοποιεί ένα αρχείο με τον κώδικα ROT13

```
[30]: def rot13(s):
    encs = ''
    for c in s:
        letter = ord(c)
        if 'a' <= c.lower() <= 'm':
            encs += chr(letter+13)
        elif 'n' <= c.lower() <= 'z':
            encs += chr(letter-13)
        else:
```

```
        encs += c
    return encs
```

```
[31]: rot13('Python 3.8')
```

```
[31]: 'Clguba 3.8'
```

```
[32]: rot13(rot13('Python 3.8'))
```

```
[32]: 'Python 3.8'
```

```
[33]: def main():
        infile = input('File to encode? ')
        outfile = input('Encoded file? ')
        try:
            fin = open(infile, 'r')
            fout = open(outfile, 'w')
        except OSError as err:
            print(err)
        else:
            for line in fin:
                fout.write(rot13(line))
        finally:
            fin.close()
            fout.close()
```

```
[34]: main()
```

```
File to encode?  grail.txt
Encoded file?  grail-enc.txt
```

```
[35]: print(open('grail-enc.txt').read())
```

```
Jr'er Xavtugf bs gur Ebhaq Gnoyr
Jr qnapr jurar're jr'er noyr
Jr qb ebhgvarf naq pubehf fprarf
Jvgu sbbgjbex vzcrppnoyr
Jr qvar jryy urer va Pnzrybg
Jr rng unz naq wnz naq Fcnz n ybg.
      -- Zbagl Clguba
```