

Σημειώσεις

[Προηγούμενο](#)[Επόμενο](#)

Αρχεία

Τα δεδομένα τα οποία χρειάζεται ένα πρόγραμμα μπορούν να διαβαστούν από ένα αρχείο. Επίσης τα αποτελέσματα τα οποία παράγονται μπορούν να καταγραφούν (να εκτυπωθούν) σε αρχείο. Ένα αρχείο μπορεί να συνδεθεί με το πρόγραμμά μας μέσω της εντολής `open`.

```
fh = open('filename.dat', 'w')
```

Η παραπάνω εντολή δημιουργεί αρχείο με όνομα `filename.dat` με σκοπό να γραφούν ('w' - write) στο αρχείο νέα αποτελέσματα. Το πρόγραμμά μας έχει πρόσβαση στο αρχείο μέσω ενός νέου αντικειμένου της `pythhon` στο οποίο δώσαμε το όνομα `fh`. Από εδώ και κάτω μπορούμε να επεμβαίνουμε στο αρχείο (με το όνομα `filename.dat`) χρησιμοποιώντας την μεταβλητή `fh`.

Παράδειγμα. Γράψτε μία σειρά ακεραίων καθώς και το τετράγωνό τους και την τετραγωνική τους ρίζα σε ένα αρχείο.

```
import math
fh = open('test.dat', 'w')
for i in range(10):
    s = '{:4d}  {:5d}   {:5.3f}\n'.format(i, i**2, math.sqrt(i))
    fh.write(s)
fh.close()
```

Ακολουθήσαμε τα εξής βήματα:

- Συνδέουμε το αρχείο (`open`) δηλώνοντας ότι θα γράψουμε σε αυτό ('w' - write).
- Δημιουργούμε μία συμβολοσειρά `s` η οποία περιέχει τη μορφή την οποία θέλουμε να εκτυπώσουμε.
- Η μέθοδος `write` γράφει στο αρχείο τη συμβολοσειρά `s`.

- Η μέθοδος `close` τερματίζει τη σύνδεση του αρχείου με το πρόγραμμα (επίσης φέρνει το αρχείο στην τελική του μορφή).

Θα δούμε πώς μπορούμε να γράψουμε δεδομένα (αποτελέσματα) σε ένα αρχείο.

Παράδειγμα. (μέθοδος `read`) Διαβάστε το περιεχόμενο ενός αρχείου και εκτυπώστε το στην οθόνη.

```
fh = open('test.dat', 'r')
s = fh.read()
fh.close()
print(s)
```

Ακολουθήσαμε τα εξής βήματα:

- Συνδέουμε το αρχείο (`open`) δηλώνοντας ότι θα διαβάσουμε από αυτό (`'r'` - `read`).
- Διαβάζουμε όλο το αρχείο (`fh.read()`) και το περιεχόμενο πηγαίνει στη μεταβλητή `s`
- Η μέθοδος `close` τερματίζει τη σύνδεση του αρχείου με το πρόγραμμα.
- Τέλος γράφουμε το περιεχόμενο του αρχείου στην οθόνη.

Ένα αρχείο το οποίο πιθανόν να είναι πολύ μεγάλο, είναι οποσδήποτε προτιμότερο να διαβαστεί κατά τμήματα. Η μέθοδος `readline()` διαβάζει μία γραμμή από αρχείο.

Παράδειγμα. (μέθοδος `readline`) Διαβάστε το περιεχόμενο ενός αρχείου κατά γραμμές και εκτυπώστε τες στην οθόνη.

```
fh = open('test.dat', 'r')
while True:
    s = fh.readline()
    if s == '': break
    print(s, end='')    # είτε: print(s[:-1]), print(s.rstrip())
fh.close()
```

Ακολουθήσαμε τα εξής βήματα:

- Συνδέουμε το αρχείο (`open`) δηλώνοντας ότι θα διαβάσουμε από αυτό.

- Αρχίζουμε μία ανακύκλωση (`while`) διαβάζοντας διαδοχικές γραμμές με τη μέθοδο `readline()` και το περιεχόμενο πηγαίνει στη μεταβλητή `s` η οποία είναι συμβολοσειρά.
- Μετά την τελευταία γραμμή του αρχείου η `readline` θα δώσει μία κενή συμβολοσειρά (συμβολοσειρά μηδενικού μήκους). Αυτό το εκμεταλλευόμαστε ως κριτήριο για να λήξει η ανακύκλωση.
- Τυπώνουμε τη γραμμή την οποία διαβάσαμε. Όμως, κάθε γραμμή που διαβάζουμε τελειώνει με τον χαρακτήρα `\n`. Για να αποφύγουμε την αλλαγή γραμμής σε κάθε εκτύπωση του `s` δίνουμε στην εντολή `print` ως τέλικό χαρακτήρα την συμβολοσειρά μηδενικού μήκους (`end = ''`).
- Τερματίζουμε τη σύνδεση του αρχείου με το πρόγραμμα.

Ερώτηση. Τι θα συμβεί με το παραπάνω πρόγραμμα αν το αρχείο περιέχει σε κάποιο σημείο μία κενή γραμμή;

Παράδειγμα. (Ανακύκλωση στο αντικείμενο του αρχείου) Υπάρχει μία ακόμα μέθοδος για να διαβάσουμε ένα αρχείο χρησιμοποιώντας απευθείας το αντικείμενο του αρχείου. Μπορούμε να διαβάσουμε το περιεχόμενο ενός αρχείου κατά γραμμές.

```
fh = open('test.dat', 'r')
for line in fh:
    print(line, end='')
fh.close()
```

Ακολουθήσαμε τα εξής βήματα:

- Συνδέουμε το αρχείο (`open`).
- Κάνουμε μία ανακύκλωση (`for`) στις διαδοχικές γραμμές του αρχείου διατρέχοντας απευθείας το αντικείμενο του αρχείου `fh`.
- Τυπώνουμε τη γραμμή την οποία διαβάσαμε αποφεύγοντας την αλλαγή γραμμής.
- Τερματίζουμε τη σύνδεση του αρχείου με το πρόγραμμα.

Ερώτηση. (Μέθοδος `readlines`) Μπορούμε να διαβάσουμε το περιεχόμενο αρχείου με τη μέθοδο `readlines` Εκτελέστε τον ακόλουθο κώδικα και δείτε σε ποιά μορφή εισάγεται το περιεχόμενο του αρχείου στην `L`.

```
fh = open('test.dat', 'r')
L = fh.readlines()
fh.close()
```

```
for e in L:  
    print(e,end="")
```

Προσάρτηση δεδομένων. Μπορούμε να ανοίξουμε ένα αρχείο για προσάρτηση (append) δεδομένων:

```
fh = open('test.dat','a')
```

Ανάγνωση και αποθήκευση. Μπορούμε να ανοίξουμε ένα αρχείο για ανάγνωση και για αποθήκευση δεδομένων:

```
fh = open('test.dat','r+')
```

Άσκηση. Έστω το αρχείο το οποίο δημιουργείται με τον κώδικα:

```
import math  
fh = open('test.dat','w')  
for i in range(10):  
    s = '{:4d},    {:5d},    {:5.3f}\n'.format(i,i**2,math.sqrt(i))  
    fh.write(s)  
fh.close()
```

Διαβάστε το αρχείο και εκχωρήστε σε μεταβλητές κάθε έναν από τους τρεις αριθμούς σε μία γραμμή του αρχείου. Ακολουθώντας ελέγξτε αν πραγματικά το 2ος αριθμός είναι το τετράγωνο του πρώτου και ο τρίτος είναι η τετραγωνική του ρίζα.

```
import math  
fh = open('test.dat','r')  
for line in fh:  
    line = line.split(',')  
    i = int(line[0])  
    i2 = int(line[1])  
    isq = float (line[2])  
    if i2 != i**2:
```

```

    print('line {:3d}:  {:4d} != {:3d}**2'.format(i,i2,i))
elif abs(isq-math.sqrt(i))>1e-3:
    print('line {:3d}:  {:f} != sqrt({:3d})'.format(i,isq,i))
else:
    print(i,i2,isq)
fh.close()

```

Παράδειγμα. Προγραμματίστε το παιχνίδι <<κρεμάλα>> το οποίο δόθηκε σε προηγούμενο μάθημα.

Παράδειγμα. (Δείτε στο [Illustrating Python via Bioinformatics Examples](#)", Κεφ. 1.7) Θα κατεβάσουμε από το διαδίκτυο ένα αρχείο με την ακολουθία DNA ενός μικροοργανισμού και θα αποθηκεύσουμε το περιεχόμενό του σε ένα αρχείο.

Ο ακόλουθος κώδικας κατεβάζει τα δεδομένα ενός αρχείου στο διαδίκτυο και τα βάζει στην μεταβλητή `data`.

```

import urllib.request                # we need the library urllib
url = 'http://hplgit.github.com/bioinf-py/data/yeast_chr1.txt'
response = urllib.request.urlopen(url) # open specific URL
data = response.read()               # reads bytes
text = data.decode('utf-8')          # turns byte data to str

```

[Δείτε και έναν πιο λεπτομερή [κώδικα](#).]

Γράφουμε τα δεδομένα της μεταβλητής `text` σε ένα αρχείο, ώστε να τα έχουμε αποθηκευμένα στον υπολογιστή μας για μελλοντική επεξεργασία.

```

fh = open("yeast_chr1.txt", "w")
fh.write(text)
fh.close()

```

Παράδειγμα. (Συνέχεια προηγούμενου) Θα επεξεργαστούμε την ακολουθία DNA που αποθηκεύσαμε στο προηγούμενο παράδειγμα και θα μετρήσουμε την συχνότητα των βάσεων A, T, G, C που περιέχονται.

Διαβάζουμε σε μία συμβολοσειρά `dna` το περιεχόμενο του αρχείου.

```
fh = open("yeast.txt", "r")
dna = fh.read()
```

Γράφουμε μία συνάρτηση η οποία βρίσκει την συχνότητα εμφάνισης των βάσεων A, T, G, C σε μία ακολουθία. Χρησιμοποιεί ένα λεξικό το οποίο έχει τέσσερες εγγραφές, μία για κάθε βάση. Δίνουμε το πλήρες πρόγραμμα το οποίο χρησιμοποιεί την συνάρτηση και τυπώνει την συχνότητα που εμφανίζεται κάθε βάση στην ακολουθία DNA.

```
def get_base_counts(dna):
    counts = {'A': 0, 'T': 0, 'G': 0, 'C': 0}
    for base in dna:
        if base in 'ATGC':
            counts[base] += 1
    return counts

fh = open("yeast_chrl.txt", "r")      # read DNA sequence from file
dna = fh.read()

counts = get_base_counts(dna)        # call function to count
occurrences of bases

dnaLength = 0                        # count total length of DNA
for base in counts:
    dnaLength += counts[base]
print("Total DNA bases:", dnaLength)

for base in counts:                  # print frequency of bases
    occurrences
    print("Frequency for base", base, counts[base]/dnaLength)
```

Σύνοψη. Συνοψίζουμε τις σημαντικότερες λειτουργίες για την επεξεργασία αρχείων. Υποθέτουμε ότι `fh` είναι μια λαβή αρχείου και `filename` είναι μια ακολουθία χαρακτήρων η οποία περιέχει το όνομα του αρχείου το οποίο επεξεργαζόμαστε:

- `open(filename, 'w')`: δημιουργεί το αρχείο `filename` (αν δεν υπάρχει ήδη) και το προετοιμάζει για να δεχθεί καινούργια στοιχεία. Επιστρέφει μια λαβή στο αρχείο.

- `open(filename, 'r')`: ανοίγει το αρχείο `filename` (αν υπάρχει) και το προετοιμάζει για ανάγνωση στοιχείων. Επιστρέφει μια λαβή στο αρχείο.
- `open(filename, 'a')`: ανοίγει το αρχείο `filename` και το προετοιμάζει για να δεχθεί καινούργια στοιχεία. Επιστρέφει μια λαβή στο αρχείο.
- `fh.read()`: επιστρέφει μια ακολουθία χαρακτήρων με τα περιεχόμενα του αρχείου με λαβή `fh`.
- `fh.readline()`: επιστρέφει ως ακολουθία χαρακτήρων την επόμενη γραμμή του αρχείου με λαβή `fh`.
- `fh.readlines()`: επιστρέφει μια λίστα στοιχεία της οποίας είναι οι γραμμές του αρχείου με λαβή `fh`.
- `fh.write(s)`: γράφει την ακολουθία χαρακτήρων `s` στο αρχείο με λαβή `fh`.
- `fh.writelines(seq)`: γράφει κάθε στοιχείο της ακολουθίας `seq` στο αρχείο με λαβή `fh`.
- `fh.close()`: κλείνει το αρχείο με λαβή `fh`.

Μελέτη

Βιβλιογραφία

1. [Python documentation](#).
2. Σημειώσεις Μ. Πλεξουσάκη.
3. J.V. Guttag, *Υπολογισμοί και προγραμματισμός με την python*.
4. Δημήτριος Καρολίδης, *Μαθαίνετε εύκολα python* (Εκδόσεις Καρολίδη, 2016).
5. Κ. Μαγκούτης, Χ. Νικολάου, *Εισαγωγή στον αντικειμενοστραφή προγραμματισμό με Python*, (Αποθετήριο "Κάλλιπος", 2016).

[Προηγούμενο](#)

[Επόμενο](#)