

ΔΙΑΧΕΙΡΙΣΗ ΑΡΧΕΙΩΝ

Κ. Φλώρος

Εισαγωγή

Γενικά, στα προγράμματα που δημιουργούμε τα δεδομένα εισόδου και εξόδου χάνονται όταν τελειώσει η εκτέλεση του προγράμματος. Χρησιμοποιούμε τα αρχεία σε πολλές περιπτώσεις όπου θέλουμε τα δεδομένα να μη χάνονται μετά το τέλος της εκτέλεσης του προγράμματος, αλλά να διαβάζουμε τα δεδομένα από ένα αρχείο του υπολογιστή στο οποίο βρίσκονται αποθηκευμένα και να γράφουμε ένα αποτέλεσμα στο ίδιο ή εναλλακτικά σε άλλο αρχείο. Αυτή η διεργασία ανάγνωσης και εγγραφής, ονομάζεται **είσοδος/έξοδος αρχείου**.

Αρχεία δεδομένων

Ως αρχεία δεδομένων χρησιμοποιούνται συνήθως αρχεία κειμένου. Ένα **αρχείο κειμένου** είναι ένα αρχείο το οποίο περιέχει μια ακολουθία χαρακτήρων και βρίσκεται αποθηκευμένο σε ένα μέσο μόνιμης αποθήκευσης, όπως ο σκληρός δίσκος.

Μπορούμε να δημιουργήσουμε ένα αρχείο δεδομένων είτε χρησιμοποιώντας ένα συντάκτη (editor), όπως το notepad είτε μέσα από την Python με κατάλληλες εντολές-συναρτήσεις.

Άνοιγμα Αρχείων

Ένα αρχείο για να χρησιμοποιηθεί, πρέπει να το ανοίξουμε με την ενσωματωμένη συνάρτηση **open()**.

Η συνάρτηση **open()** μας επιστρέφει ένα αντικείμενο του αρχείου και μπορούμε να το χρησιμοποιήσουμε για να εκτελέσουμε διάφορες λειτουργίες σε αυτό.

Η σύνταξή της είναι:

Όνομα-Αντικειμένου=**open** (“όνομα_αρχείου”, “τρόπος προσπέλασης”)

open (“όνομα_αρχείου”, “τρόπος προσπέλασης”)

Η συνάρτηση **open()** δέχεται δυο ορίσματα:

το πρώτο είναι το όνομα του αρχείου, με το οποίο το αναγνωρίζει το λειτουργικό σύστημα.

Το δεύτερο είναι ένα ειδικό σύμβολο (σημαία - flag) που καθορίζει τον τρόπο προσπέλασης του αρχείου.

Ορίσματα τρόπου προσπέλασης του αρχείου	Λειτουργία
“r”	Ανάγνωση
“w”	Εγγραφή (διαγραφή προηγούμενων περιεχομένων, αν υπάρχουν)
“a”	Προσθήκη (append) (διατήρηση προηγούμενων περιεχομένων)
“r+”	Άνοιγμα αρχείου και για ανάγνωση και για εγγραφή
“b”	Αρχείο δυαδικής μορφής

Άνοιγμα Αρχείων - Παράδειγμα

Π.χ. `keimeno=open('ergasia.txt')`.

Στο αντικείμενο κείμενο αντιστοιχεί το αρχείο `ergasia.txt`. Με το όνομα αυτό θα αναφερόμαστε στο αρχείο μέσα στον κώδικά μας.

Σημείωση: Όταν δεν χρησιμοποιείται το δεύτερο όρισμα , τότε θεωρείται εξ ορισμού ότι είναι το “r”.

Άνοιγμα Αρχείων – Παραδείγματα 2

Π.χ. `open("ergasia.txt","w")`.

Η συνάρτηση αυτή, αν δεν υπάρχει το αρχείο `ergasia.txt`, τότε το δημιουργεί, ενώ αν υπάρχει, τότε θα έχουμε ως αποτέλεσμα να χαθούν τα περιεχόμενά του.

Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση με το όρισμα `"a"`

Π.χ. `open("ergasia.txt", "a")`,
η οποία, αν δεν υπάρχει το αρχείο, το δημιουργεί, ενώ, αν υπάρχει, το ανοίγει σε κατάσταση προσθήκης δεδομένων στο τέλος του.

Κλείσιμο Αρχείων

Όταν ολοκληρώσουμε τις λειτουργίες που θέλουμε να εκτελεστούν στο αρχείο, καλούμε απαραίτητα τη συνάρτηση **close()**. Αυτό δηλώνει ότι τελειώσαμε με τη χρήση του και οδηγεί το Λειτουργικό Σύστημα στην αποθήκευση δεδομένων που βρίσκονται ακόμη στη μνήμη.

Π.χ. `keimeno.close()`

Η **close** κλείνει το αρχείο, παρόμοια με την επιλογή File→Save στον επεξεργαστή κειμένου.

Για να ελέγξουμε, αν ένα αρχείο όντως έκλεισε, πρέπει να χρησιμοποιήσουμε την ιδιότητα (attribute) `closed`, η οποία επιστρέφει την τιμή `True` ή `False`, ανάλογα, αν είναι κλειστό ή ανοιχτό το αρχείο.

Κλείσιμο Αρχείων - Παράδειγμα

- >>> fin = open("workfile.txt")
- >>> fin.closed
- # False
- >>> fin.close()
- >>> fin.closed
- # True

Εγγραφή σε αρχείο

Έστω το αρχείο “programs.txt” με τα ακόλουθα περιεχόμενα:

1. Γλώσσα προγραμματισμού Python
2. Γλώσσα προγραμματισμού java

Για να προσθέσουμε περιεχόμενα σ' αυτό το αρχείο πρέπει πρώτα να το ανοίξουμε με την εντολή:

```
arχειο=open("programs.txt","a")
```

Τι θα γίνει αν το άνοιγα με την εντολή

```
arχειο=open("programs.txt","w");
```

Μέθοδος write()

Για την εγγραφή δεδομένων σε αρχείο χρησιμοποιούμε τη μέθοδο **write()** και ως όρισμα τη συμβολοσειρά που θέλουμε να εισάγουμε στο αρχείο. Σύνταξη: **ον-αντικ-αρχείου.write(συμβολοσειρά)**.

Για παράδειγμα:

```
>>> arxheio = open("programs.txt", "a")
```

```
>>> arxheio.write("3. Γλώσσα προγραμματισμού C\n")
```

Το αρχείο "programs.txt" έχει τώρα τη μορφή:

1. Γλώσσα προγραμματισμού Python
2. Γλώσσα προγραμματισμού java
3. Γλώσσα προγραμματισμού C

Σημείωση - write()

Σε ένα αρχείο κειμένου, αν θέλουμε να υπάρχουν γραμμές και όχι συνεχόμενοι χαρακτήρες, πρέπει να σημειώνουμε την αλλαγή γραμμής με το χαρακτήρα “\n”.

Το όρισμα της μεθόδου write πρέπει να είναι συμβολοσειρά. Αν θέλουμε να εισάγουμε κάτι διαφορετικό, όπως για παράδειγμα έναν αριθμό, ο ευκολότερος τρόπος είναι, να μετατρέψουμε τον αριθμό σε συμβολοσειρά, με χρήση της συνάρτησης str.

```
>>> x = 9
```

```
>>> keimeno.write(str(x))
```

Ανάγνωση περιεχομένων αρχείου

Οι πιο διαδεδομένες μέθοδοι για διάβασμα των περιεχομένων ενός αρχείου, είναι η **read()** και η **readline()**.

Μέθοδος **read()**

Η **read()** διαβάζει ένα πλήθος χαρακτήρων από την αρχή του αρχείου.

Σύνταξη: **ον-αντικειμ-αρχείου.read([count]);**

Εδώ, η παράμετρος **count** καθορίζει τον αριθμό των χαρακτήρων που θα διαβασθούν από το αρχείο, ξεκινώντας την ανάγνωση από την αρχή του μέχρι το πλήθος των χαρακτήρων που ορίζονται με την **count**. Στην περίπτωση που η παράμετρος **count** λείπει, τότε γίνεται ανάγνωση μέχρι να διαβαστεί η ένδειξη τέλους του αρχείου.

Ανάγνωση περιεχομένων αρχείου-Παραδείγματα

Παράδειγμα 1: >>> keimeno = open('programs.txt', 'r')

>>> print keimeno.read(1)

1

Παράδειγμα 2: Διάβασμα και εμφάνιση των επόμενων 10 χαρακτήρων

>>> print keimeno.read(10)

. Γλώσσα π

Παράδειγμα 3: Εμφάνιση ολόκληρου του υπόλοιπου αρχείου

>>> print keimeno.read()

ρογραμματισμού Python

2. Γλώσσα προγραμματισμού java

3. Γλώσσα προγραμματισμού C

Η μέθοδος readline()

Η **readline** διαβάζει μια γραμμή του αρχείου, δηλαδή διαβάζει διαδοχικούς χαρακτήρες από ένα αρχείο μέχρι να συναντήσει το χαρακτήρα νέας γραμμής και επιστρέφει το αποτέλεσμα:

Παράδειγμα

```
>>> keimeno = open("programs.txt")
```

```
>>> print keimeno.readline()
```

1. Γλώσσα προγραμματισμού Python

Το keimeno καταγράφει τη θέση που βρίσκεται μέσα στο αρχείο και έτσι αν ξανακαλέσουμε τη readline θα πάρουμε την επόμενη γραμμή.

Τι θα εμφανίσει η εντολή;

```
>>> print keimeno.readline()
```

Παράδειγμα

Να γραφεί πρόγραμμα που να τυπώνει όλες τις γραμμές του αρχείου programs.txt.

```
arχειο=open("programs.txt",'r')
```

```
for line in arχειο:
```

```
    print line,
```

Εντοπισμός θέσης στο αρχείο

Η μέθοδος **tell()** επιστρέφει έναν ακέραιο που περιέχει την τρέχουσα θέση στο αρχείο, υπολογισμένη σε χαρακτήρες (bytes) από την αρχή του αρχείου. Με άλλα λόγια, η επόμενη ανάγνωση ή εγγραφή θα γίνει σε εκείνη τη θέση του αρχείου.

Σύνταξη : **ον-αντικειμ-αρχείου.tell()**

Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση του παρακάτω τμήματος προγράμματος;

```
arxeio=open("programs.txt", "r")
```

```
arxeio.read(20)
```

```
print arxeio.tell()
```

Αλλαγή τρέχουσας θέσης του αρχείου

Για να αλλάξουμε την τρέχουσα θέση του αρχείου, μπορούμε να χρησιμοποιήσουμε την μέθοδο **seek()**.

Σύνταξη : **ον-αντικειμ-αρχείου.seek (offset[, from_what])**

Η θέση υπολογίζεται προσθέτοντας offset (πλήθος bytes) σε ένα σημείο αναφοράς, το οποίο επιλέγεται από το from_what όρισμα.

Αν το from_what έχει τιμή 0 μετρά από την αρχή του αρχείου

Αν έχει τιμή 1, χρησιμοποιεί την τρέχουσα θέση του αρχείου

και αν έχει τιμή 2, χρησιμοποιεί το τέλος του αρχείου.

Αλλαγή τρέχουσας θέσης του αρχείου-Παράδειγμα

```
>>> arxerio = open("workfile.txt", "r+")
```

```
>>> arxerio.write("0123456789abcdef")
```

```
>>> arxerio.seek(5)          # πηγαίνει στο 6ο byte στο αρχείο
```

```
>>> arxerio.read(1)
```

```
'5'
```

```
>>> arxerio.seek(-3, 2)      # πηγαίνει στο 3ο byte πριν το τέλος
```

```
>>> arxerio.read(1)
```

```
'd'
```

Δραστηριότητα 1

Ας πειραματιστούμε γράφοντας κώδικα σε Python, ο οποίος διαβάζει το αρχείο **"input_file.txt"** και το ξαναγράφει στο νέο αρχείο **"output_file.txt"**, όπου πριν από κάθε γραμμή θα προσθέτει τον αύξοντα αριθμό της.

- Έστω τα δύο αρχεία **inputfile** και **outputFile**.
- Ανοίγω για διάβασμα το **inputfile** και για γράψιμο το αρχείο **outputFile**
- Δίνω αρχική τιμή στο αύξοντα αριθμό της γραμμής **linecounter = 1**
- Για κάθε γραμμή του αρχείου **inputfile** :
 - Γράφω στο αρχείο **outputFile** τον αύξοντα αριθμό και την γραμμή αυτή.
 - Αυξάνω κατά 1 τον αύξοντα αριθμό **linecounter**.
- Πάντα στο τέλος κλείνω τα αρχεία που άνοιξα.

Δραστηριότητα 1 - Λύση

```
inputfile = open("input_file.txt","r")  
outputFile = open("output_file.txt","w")  
linecounter = 1  
for line in inputfile:  
    outputFile.write(str(linecounter)+". "+line)  
    linecounter = linecounter + 1  
inputfile.close()  
outputFile.close()
```

Δραστηριότητα 2

Να γράψετε ένα πρόγραμμα το οποίο θα ενώνει δυο αρχεία κειμένου σε ένα τοποθετώντας τα περιεχόμενα του δεύτερου αρχείου μετά από αυτά του πρώτου.

- Έστω τα δύο αρχεία **arxeio1** και **arxeio2**.
- Ανοίγω για διάβασμα το **arxeio2** και με τη **read** μεταφέρω το περιεχόμενό του σε μία μεταβλητή **s**.
- Ανοίγω για προσθήκη ('**a**') το **arxeio1** και με την **append** προσθέτο το **s**.
- Πάντα στο τέλος κλείνω τα αρχεία που άνοιξα.

Δραστηριότητα 2 - Λύση

```
f1 = open('arxeio1.txt', 'a')  
f2 = open('arxeio2.txt', 'r')  
s = f2.read()  
f1.write(s)  
f1.close()  
f2.close()
```

Άσκηση 1

Να γράψετε πρόγραμμα σε Python το οποίο:

- **Θα δημιουργεί μια λίστα με όλους τους μονούς αριθμούς από το 1 μέχρι το 999**
- **και θα γράφει αυτούς τους αριθμούς σε ένα αρχείο έναν σε κάθε γραμμή.**
- **Στη συνέχεια θα διαβάζει τους αριθμούς από το αρχείο και**
- **θα γράφει το άθροισμά τους στο τέλος του αρχείου χωρίς να διαγράψει τα περιεχόμενα του αρχείου.**

Άσκηση 1 -Λύση

```
# δημιουργία λίστας μονών αριθμών  
evenL = range(1,1000,2)
```

```
# και αποθήκευσή τους σε αρχείο  
f = open('odd_list.txt', 'w')  
for num in evenL:  
    f.write( str(num) + '\n' )  
f.close()
```

```
# άνοιγμα αρχείου για ανάγνωση  
f = open('odd_list.txt', 'r')  
s = 0  
for line in f:  
    s= s + int(line)  
f.close()
```

```
# άνοιγμα αρχείου για προσθήκη αθροίσματος  
f = open('odd_list.txt', 'a')  
f.write( str(s) + '\n')  
f.close()
```

Άσκηση 2

Να γράψετε πρόγραμμα σε python το οποίο:

- Θα δημιουργεί δύο λίστες POLI και THERMO,
- θα τις γεμίζει με 10 πόλεις και
- τις αντίστοιχες θερμοκρασίες που θα εισάγει ο χρήστης.
- Κατόπιν με τη μέθοδο bubble sort
- θα ταξινομεί με φθίνουσα σειρά τις λίστες ως προς τις θερμοκρασίες.
- Τέλος θα δημιουργεί αρχείο με όνομα "neo.txt"
- όπου σε κάθε γραμμή θα προσθέτει τον αύξοντα αριθμό της,
- το στοιχείο της ταξινομημένης λίστας POLI και το αντίστοιχο στοιχείο της λίστας THERMO

Άσκηση 2 - Λύση

```
THERMO=[ ]
POLI=[ ]
for i in range(0,10):
    num=int(input('dose thermokrasia: '))
    THERMO.append(num)
    city=raw_input('dose poli: ')
    POLI.append(city)
print "Η αταξινόμητες λίστες είναι: "
for i in range(0,10):
    print POLI[i],THERMO[i]
```

Άσκηση 2 – Λύση (συνέχεια)

```
N = len(THERMO)
for i in range(N-1):
    for j in range(N-1,i,-1):
        if THERMO[j] > THERMO[j-1]:
            THERMO[j],THERMO[j-1]=THERMO[j-1],THERMO[j]
            POLI[j],POLI[j-1]=POLI[j-1],POLI[j]

print "Η ταξινομημένες λίστες είναι: "
for i in range(0,10):
    print POLI[i],THERMO[i]

outputFile = open("new.txt", "w")
for i in range(0,10):
    outputFile.write(str(i+1) + ". " + POLI[i] + " "+ str(THERMO[i])+ "\n")
outputFile.close()
```

Άσκηση 3

- Δίνεται αρχείο "rth.txt" το οποίο περιέχει ονόματα πόλεων και τη μέγιστη θερμοκρασία τους σε μία συγκεκριμένη ημέρα. Το αρχείο έχει την παρακάτω δομή (ενδεικτικά):

Πάτρα

25

Λάρισα

27

...

- Συνεπώς οι γραμμές που βρίσκονται σε περιττές θέσεις περιέχουν ονόματα πόλεων και οι γραμμές που βρίσκονται σε άρτιες θέσεις περιέχουν σε ακέραιο αριθμό τη μέγιστη θερμοκρασία της πόλης.
- Να αναπτύξετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Άσκηση 3 -συνέχεια

- 1) Να διαβάσει το αρχείο "rth.txt" γραμμή προς γραμμή και ανάλογα αν η γραμμή είναι σε περιττή ή άρτια θέση, να καταχωρεί τα ονόματα των πόλεων σε μία λίστα POL και τις αντίστοιχες θερμοκρασίες τους σε μία λίστα THER.
- 2) Να υπολογίζει και να εμφανίζει το μέσο όρο των θερμοκρασιών όλων των πόλεων.
- 3) Να ταξινομεί τη λίστα THER με χρήση του αλγορίθμου ταξινόμησης της ευθείας ανταλλαγής (φουσαλίδα-bubble sort) σε φθίνουσα σειρά ως προς τις θερμοκρασίες αναδιατάσσοντας συγχρόνως τη λίστα POL έτσι ώστε να διατηρείται η αντιστοίχιση πόλεων-θερμοκρασιών.
- 4) Θεωρώντας ότι μπορεί να υπάρχουν περισσότερες από μία πόλεις με την ίδια θερμοκρασία, να εμφανίζει την υψηλότερη θερμοκρασία που έχει καταχωρηθεί και τα ονόματα των πόλεων που έχουν αυτή τη θερμοκρασία.

Άσκηση 3 - Λύση

```
POL=[ ]
THER=[ ]
fin=open("pth.txt","r")
m=0
for line in fin:
    m=m+1
    if m%2==1:
        POL.append(line)
    else:
        THER.append(int(line))
fin.close()

sum = 0.0
for num in THER:
    sum = sum + int(num)
mo = sum/len(THER)
print mo
```

Άσκηση 3 – Λύση (συνέχεια)

```
N = len(THER)
for i in range( N-1 ):
    for j in range( N-1 , i , -1 ):
        if THER [ j ] > THER [ j-1 ]:
            THER [ j ] , THER [ j-1 ] = THER [ j-1 ] , THER [ j ]
            POL [ j ] , POL [ j-1 ] = POL [ j-1 ] , POL [ j ]

maxi=THER[0]
print "Η υψηλότερη θερμοκρασία είναι: ",maxi
for i in range(N):
    if THER[i]==maxi:
        print POL[i]
```