

Python – 6. Διαχείριση Αρχείων

Διαχείριση Αρχείων

Χρησιμοποιούμε δύο τύπους αρχείων: αυτά που περιέχουν το πρόγραμμα που θα εκτελέσουμε και εκείνα που περιέχουν τα δεδομένα, τα οποία, όταν εκτελεστεί το πρόγραμμα, τα διαβάζει και πιθανά τα ενημερώνει.

Ως αρχεία δεδομένων χρησιμοποιούνται συνήθως αρχεία κειμένου (.txt).

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

Δημιουργία - Άνοιγμα - Κλείσιμο Αρχείων

ΑΝΟΙΓΜΑ αρχείου

```
open ("όνομα_αρχείου", "τρόπος προσπέλασης")
```

Τρόποι Προσπέλασης:

Αν παραλειφθεί, προεπιλεγμένη τιμή είναι το "r".

Παράδειγμα

```
>>> fin = open('words.txt')
>>> print fin
```

"r"	Ανάγνωση (read)
"w"	Εγγραφή (write) - διαγραφή προηγούμενων περιεχομένων - Χρησιμοποιείται για τη δημιουργία αρχείου
"a"	Προσθήκη (append) - διατήρηση προηγούμενων περιεχομένων
"r+"	Ανάγνωση και Εγγραφή
"b"	Αρχείο δυαδικής μορφής

ΔΗΜΙΟΥΡΓΙΑ αρχείου

```
open ("όνομα_αρχείου", "w")
```

```
open ("όνομα_αρχείου", "a")
```

Παραδείγματα

```
f = open("words.txt", "w")
```

Αν το αρχείο words.txt δεν υπάρχει τότε δημιουργείται ενώ αν υπάρχει, χάνεται το περιεχόμενό του.

```
f = open("words.txt", "a")
```

Αν το αρχείο words.txt δεν υπάρχει τότε δημιουργείται ενώ αν υπάρχει, μπορούμε να προσθέσουμε δεδομένα στο τέλος του.

ΜΕΘΟΔΟΙ διαχείρισης αρχείων

```
read()
```

```
readline()
```

```
write()
```

```
close()
```

```
tell()
```

```
seek()
```

```
closed
```

seek (Θ, Σ)

όπου Σ λαμβάνει τιμές 0,1,2

Αν Σ=0, πάει στη θέση Θ μετρώντας από την αρχή του αρχείου

Αν Σ=1, προχωράει Θ θέσεις ξεκινώντας από την τρέχουσα θέση

Αν Σ=2, πάει στη θέση Θ μετρώντας από το τέλος του αρχείου

Αν δεν προσδιοριστεί, θεωρείται ότι είναι Σ=1.

Παράδειγμα 1ο

```
>>> f = open("test.txt", "w")
>>> f.write('1st line.\n')
>>> f.write('2nd line.\n')
>>> f.close()
```

Δημιουργήθηκε το αρχείο test.txt που περιέχει δύο γραμμές:

```
1st line.
2nd line.
```

Παράδειγμα 2ο

```
>>> f=open("test.txt")
>>> print f.readline()
1st line.
>>> print f.readline()
2nd line.
>>> f.close()
```

Παράδειγμα 3ο

```
>>> f=open("test.txt")
>>> print f.read()
1st line.
2nd line.
>>> f.close()
```

Παράδειγμα 4ο

```
>>> f=open('test.txt', 'a')
>>> a=43
>>> s=str(a)
>>> f.append(s)
>>> f.write(s)
>>> f.close()
```

Παράδειγμα 5ο

```
>>> f=open('test.txt')
>>> print f.read()
1st line.
2nd line.
43
>>> f.closed
False
```

Παράδειγμα 6ο

```
>>> f=open('test.txt')
>>> print f.read(3)
1st
>>> f.close()
```

Παράδειγμα 7ο

```
>>> f=open('test.txt')
>>> print f.tell()
0
>>> f.seek(4)
>>> print f.read(5)
line.
>>> f.close()
```

Εντοπισμός θέσης στο αρχείο

Η μέθοδος **f.tell()** επιστρέφει έναν ακέραιο που περιέχει την τρέχουσα θέση στο αρχείο, υπολογισμένη σε χαρακτήρες (bytes) από την αρχή του αρχείου.

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

Η θέση υπολογίζεται προσθέτοντας Θ (πλήθος bytes) σε ένα σημείο αναφοράς, το οποίο επιλέγεται από το Σ.

Αν το Σ έχει τιμή 0, μετρά από την αρχή του αρχείου, αν έχει 1, χρησιμοποιεί την τρέχουσα θέση του αρχείου και αν έχει 2, χρησιμοποιεί το τέλος του αρχείου.

Είναι το [ENTER] χαρακτήρας;

Δημιουργήστε το αρχείο test1.txt με το παρακάτω περιεχόμενο:

```
1
2
```

όπου μετά το χαρακτήρα "1" έχει πατηθεί [ENTER] και μετά το χαρακτήρα "2" δεν έχει πατηθεί [ENTER].

Το παραπάνω αρχείο έχει δύο (2) χαρακτήρες ή παραπάνω;

Πώς μπορούμε να ελέγξουμε αυτό;

Μπορείτε να σκεφτείτε έναν τρόπο με τις μεθόδους και συναρτήσεις που ξέρουμε;

```
f=open('test1.txt') #ανοίγω το αρχείο
count=0 #μετρητής χαρακτήρων
ch="" #αρχική τιμή στο ch για να μπει στη while
while ch != "2": #επανάλαβε μέχρι το "2"
    ch=f.read(1) #διαβάζω ένα χαρακτήρα
    count += 1 #αυξάνω τον μετρητή κατά ένα
f.close() #κλείνω το αρχείο
print count #εμφανίζω πόσους χαρακτήρες διάβασα...
```

```
>>>
3
```

Από το αποτέλεσμα αυτό, το μόνο σίγουρο συμπέρασμα είναι ότι το [ENTER] είναι ένας χαρακτήρας!

```
f=open('test1.txt') #ανοίγω το αρχείο
f.seek(0,2)
count=f.tell()
f.close() #κλείνω το αρχείο
print count #εμφανίζω πόσους χαρακτήρες διάβασα...
```

```
>>>
4
```

Διαχείριση γραμμή - γραμμή

Με την εντολή:

```
f = open('test2.txt', 'r')
for line in f:
    ...
```

διασχίζουμε το αρχείο γραμμή - γραμμή

Δημιουργήστε το αρχείο test2.txt με το παρακάτω περιεχόμενο:

```
1. Γιάνν
2. Θόδωρ
3. Πηνελ
4. Μαρία
5. Ηλίας
6. Θανάσ
7. Αντών
8. Χρήστ
```

Φτιάξτε και εκτελέστε το πρόγραμμα:

```
f=open('test2.txt')
for line in f: print line[0]
f.close()
```

Δραστηριότητες**Δραστηριότητα 1**

Στο φάκελο C:\Python27:

1. Δημιουργήστε ένα φάκελο με το όνομα TxtFiles.
2. Μέσα στον παραπάνω φάκελο δημιουργήστε το αρχείο names.txt το οποίο περιέχει τα ονόματά σας, το καθένα σε διαφορετική γραμμή.
3. Φτιάξτε ένα πρόγραμμα σε python το οποίο διαβάζει το names.txt και δημιουργεί το new_names.txt που έχει αρίθμηση σε κάθε γραμμή.

Λύση

```
f1 = open("TxtFiles/names.txt", "r")
f2 = open("TxtFiles/new_names.txt", "w")
i=0
for L in f1:
    i += 1
    s = str(i) + ' ' + L
    f2.write(s)
f1.close()
f2.close()
```

Δραστηριότητα 2

Φτιάξτε ένα πρόγραμμα σε python το οποίο θα διαβάζει ένα αρχείο (π.χ. το names.txt) και θα επιστρέφει το πλήθος των γραμμών που περιέχει.

Λύση

```
f1 = open("TxtFiles/names.txt", "r")
i=0
for L in f1:
    i += 1
f1.close()
print i
```

Δραστηριότητα 3 [Δ]

Να γράψετε πρόγραμμα στη γλώσσα Python, το οποίο θα δέχεται ως είσοδο το όνομα ενός αρχείου, θα εμφανίζει το περιεχόμενό του κατά γραμμή. Στη συνέχεια, θα γράφει σε ένα άλλο αρχείο, τις γραμμές του αρχείου με την αντίστροφη σειρά.

Δραστηριότητα 4

Να γράψετε πρόγραμμα στη γλώσσα Python, το οποίο:

1. Δημιουργεί μια λίστα με τα τετράγωνα των αριθμών από το 1 έως και το 10.
2. Δημιουργεί ένα αρχείο γράφοντας τους παραπάνω αριθμούς, έναν σε κάθε γραμμή.
3. Διαβάζει από την αρχή το αρχείο και τυπώνει το περιεχόμενό του γραμμή - γραμμή.

Λύση

```
L = []
for i in range(1,11):
    num = i**2
    L.append(num)

F = open('nums.txt', 'w')
for item in L: F.write(str(item) + '\n')
F.close()

F = open('nums.txt', 'r')
for line in F: print line
F.close()
ή
L = [i**2 for i in range(1,11)]

F = open('nums.txt', 'w')
for item in L: F.write(str(item) + '\n')
F.close()

F = open('nums.txt', 'r')
for line in F: print line
F.close()
```

Δραστηριότητα 5

Να γράψετε ένα πρόγραμμα σε Python το οποίο:

1. Θα δημιουργεί μια λίστα με όλους τους άρτιους αριθμούς από το 2 μέχρι το 1000 και θα γράφει αυτούς τους αριθμούς σε ένα αρχείο έναν σε κάθε γραμμή.
2. 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.

Ανάλυση

- Την λίστα με τους άρτιους θα τη δημιουργήσουμε με τη συνάρτηση range().
- Το αρχείο θα ανοίγουμε με την open() δηλώνοντας ως τρόπο προσπέλασης το "w" όταν θέλουμε να το δημιουργήσουμε και να γράψουμε σε αυτό, το "r" όταν θέλουμε να διαβάσουμε το περιεχόμενό του και

το "a" όταν θέλουμε να προσθέσουμε κάτι στο τέλος του αρχείου.

- Για να γράφουμε στο αρχείο θα χρησιμοποιήσουμε τη μέθοδο write και για να γράψουμε ένα αριθμό σε κάθε γραμμή την str() και το '\n'.
- Για να κλείσουμε το αρχείο θα χρησιμοποιήσουμε τη μέθοδο close.
- Για να διαβάζουμε από το αρχείο γραμμή-γραμμή χρησιμοποιούμε την for.
- Για να μετατρέψουμε μια συμβολοσειρά (string) σε ακέραιο θα χρησιμοποιήσουμε την int().

Λύση

```
# δημιουργία λίστας ζυγών αριθμών
# και αποθήκευσή τους σε αρχείο
evenL = range(2,1001,2)
f = open('even_list.txt', 'w')
for number in evenL:
    f.write( str(number) + '\n' )
f.close()

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

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

Δραστηριότητα 6

Να γράψετε ένα πρόγραμμα σε Python το οποίο θα διαβάζει ένα κείμενο από ένα αρχείο και θα εμφανίζει τις λέξεις κάθε πρότασης σε αντίστροφη σειρά, δηλαδή την πρώτη λέξη τελευταία και την τελευταία πρώτη. Δεχόμαστε ότι κάθε πρόταση τελειώνει με τελεία και οι λέξεις είναι χωρισμένες με ένα κενό. Δεν υπάρχουν άλλα σημεία στίξεως.

Δίνονται οι συναρτήσεις διαχείρισης στοίβας createStack(), isEmpty(), pop() και push().

Ανάλυση

- Έστω το αρχείο keimeno.txt που περιέχει ως κείμενο την εκφώνηση.
- Ανοίγουμε το αρχείο και με ένα read περνάμε το περιεχόμενό του σε ένα string.
- Το διαβάζουμε χαρακτήρα - χαρακτήρα και όταν:
 - α) βρούμε κενό δημιουργούμε μια λέξη που τη βάζουμε σε στοίβα
 - β) βρούμε τελεία δημιουργούμε την τελευταία λέξη, την βάζουμε στην στοίβα και εμφανίζουμε τα στοιχεία της στοίβας ένα - ένα αδειάζοντας ταυτόχρονα τη στοίβα.

- Υλοποιούμε το παραπάνω με μια εξωτερική for (για κάθε γράμμα του string) και μία εσωτερική while για κάθε πρόταση του string. Είναι προφανές ότι η τελευταία της τελευταίας πρότασης θα είναι ο τελευταίος χαρακτήρας του string!

Λύση

```
def push(st,item):
    st = st + [item]
    return st

def pop(st):
    return st.pop()

def isEmpty(st):
    return len(st) == 0

def createStack():
    return []

F = open('keimeno.txt', 'r')
N = F.read()
F.close()
N = len(S)

STACK = createStack()
W = ""

for i in range(N):
    if S[i] == " ":
        STACK = push(STACK,W)
        W = ""
    elif S[i] == ".":
        STACK = push(STACK,W)
        while not isEmpty(STACK):
            print pop(STACK)
        W = ""
    else:
        W += S[i]
```

Δραστηριότητα 7

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

Ανάλυση

- Έστω τα δύο αρχεία A1 και A2.
- Ανοίγω για διάβασμα το A2 και με τη read μεταφέρω το περιεχόμενό του σε μία μεταβλητή S.
- Ανοίγω για προσθήκη ('a') το A1 και με την append προσθέτο το S.
- Πάντα στο τέλος κλείνω τα αρχεία που άνοιξα...

Λυση

```
F1 = open('a1.txt', 'a')
F2 = open('a2.txt', 'r')
S = F2.read()
F1.append(S)
F1.close()
F2.close()
```

Δραστηριότητα 8

Να υλοποιήσετε μια συνάρτηση copy(source, destination) η οποία θα δημιουργεί ένα αντίγραφο του αρχείου με όνομα source στο αρχείο με όνομα destination.

Ανάλυση

- Ανοίγω το αρχείο source για διάβασμα ('r') και δημιουργώ το destination ('w').
- Με τη read μεταφέρω το περιεχόμενο του source σε μία μεταβλητή S.
- Με την write γράφω το S μέσα στο destination.
- Πάντα στο τέλος κλείνω τα αρχεία που άνοιξα...

Λυση

```
def copy(source, destination):
    f1 = open(source, 'r')
    f2 = open(destination, 'w')
    st = f1.read()
    f2.write(st)
    f1.close()
    f2.close()

# έλεγχος
copy('keimeno.txt', 'antigrafo.txt')
f=open('keimeno.txt', 'r')
s=f.read()
print s
f.close()
f=open('antigrafo.txt', 'r')
s=f.read()
print s
f.close()
```