

ΗΜΥ01Κ06
Επιστημονικός Προγραμματισμός με Python



6+1 ερωτήσεις επανάληψης για τη Διάλεξη 08
Αρχεία

Φθινόπωρο 2025

Ερώτηση 1/7

Με ποιους 3 διαφορετικούς τρόπους μπορούμε να ανοίξουμε ένα αρχείο για εγγραφή;

- **r** (*read*) : Το αρχείο ανοίγει για *ανάγνωση* (*default – μπορεί να παραλειφθεί*) αν το αρχείο δεν υπάρχει, επιστρέφεται μήνυμα λάθους
- ➔ **w** (*write*) : Το αρχείο ανοίγει κενό για *εγγραφή* αν το αρχείο προϋπάρχει, τότε το περιεχόμενό του καθαρίζεται
- ➔ **a** (*append*) : Το αρχείο ανοίγει για *προσθήκη νέων εγγραφών στο τέλος*
- ➔ **x** (*exclusive creation*) : Το αρχείο ανοίγει κενό για *εγγραφή*, εφόσον δεν προϋπάρχει αν το αρχείο προϋπάρχει, επιστρέφεται μήνυμα λάθους

Ερώτηση 2/7

Πως διαφέρουν τα 'r+', 'x+', 'w+' και 'a+' κατά το άνοιγμα ενός αρχείου;

- 'r+' : Το αρχείο ανοίγει για διάβασμα μόνο εφόσον υπάρχει
Αφού ανοίξει κανονικά, τότε μπορούμε επιπλέον να γράφουμε σε αυτό
- 'x+' : Το αρχείο δημιουργείται για γράφισμο, μόνο εφόσον δεν υπάρχει
Αν υπήρχε ήδη αρχείο με αυτό το όνομα, τότε *εγείρεται μια εξαίρεση (σφάλμα)*
Αφού ανοίξει κανονικά, και γράψουμε κάτι μέσα του, τότε μπορούμε επιπλέον να διαβάζουμε από αυτό
- 'w+' : Το αρχείο δημιουργείται για γράφισμο σε κάθε περίπτωση
Αν υπήρχε ήδη αρχείο με αυτό το όνομα, τότε *χάνεται το υπάρχον περιεχόμενό του*
Αφού ανοίξει κανονικά, και γράψουμε κάτι μέσα του, τότε μπορούμε επιπλέον να διαβάζουμε από αυτό
- 'a+' : Το αρχείο δημιουργείται για γράφισμο σε κάθε περίπτωση
Αν υπήρχε ήδη αρχείο με αυτό το όνομα, τότε *το υπάρχον περιεχόμενό του διατηρείται*
Αφού ανοίξει κανονικά, τότε μπορούμε επιπλέον να διαβάζουμε από αυτό

Ερώτηση 3/7

Ποιες είναι οι τρεις μέθοδοι ανάγνωσης ενός αρχείου κειμένου;

f.*read*(k) #διάβασε το πολύ k χαρακτήρες

Κάθε φορά που καλείται διαβάζει και επιστρέφει σε μια συμβολοσειρά *το πολύ*^(*) k *χαρακτήρες*, ή *όλο το υπόλοιπο περιεχόμενο* του αρχείου, αν παραλειφθεί το k

f.*readline*() #διάβασε μέχρι να συναντήσεις \n

Κάθε φορά που καλείται διαβάζει και επιστρέφει τους χαρακτήρες *μέχρι και το τέλος της γραμμής* (χαρακτήρα αλλαγής γραμμής \n). Αν *δεν συναντήσει χαρακτήρα αλλαγής γραμμής*, τότε *συνεχίζει να διαβάζει μέχρι το τέλος του αρχείου*

f.*readlines*() #διάβασε σε λίστα τις υπόλοιπες γραμμές

Όταν καλείται, *διαβάζει το υπόλοιπο του αρχείου* και επιστρέφει μια λίστα *συμβολοσειρών*, καθεμία από τις οποίες αντιστοιχεί σε *μια γραμμή του αρχείου* δηλαδή τερματίζεται με τον *χαρακτήρα αλλαγής γραμμής* (\n)

Επιστρέφουν
συμβολοσειρές

Επιστρέφει λίστα
συμβολοσειρών

^(*) επιστρέφει λιγότερους από k αν δεν υπάρχουν αρκετοί να διαβάσει μέχρι το τέλος του αρχείου

Ερώτηση 4/7

Τι επιστρέφει η μέθοδος `f.read(n)` όπου `f` ένα αρχείο ανοικτό για διάβασμα;

Εξαρτάται από το αν το αρχείο είναι *κειμένου* ή *δυναμικό*

- Αν το `f` είναι *αρχείο κειμένου*, τότε η εντολή `f.read(n)` επιστρέφει μια *συμβολοσειρά* `n` (το πολύ – γιατί;) χαρακτήρων
- Αν το `f` είναι *δυναμικό αρχείο*, τότε η εντολή `f.read(n)` επιστρέφει ένα *"αντικείμενο δυναμικών δεδομένων"* (*bytes object*) μεγέθους `n` (το πολύ) bytes

Κατ' αντιστοιχία η μέθοδος `f.write(<καποιο_αντικείμενο>)` επιστρέφει τον *αριθμό των χαρακτήρων* ή τον *αριθμό των bytes* που γράφτηκαν στο αρχείο, ανάλογα αν πρόκειται για *αρχείο κειμένου*, ή *δυναμικό*

Ερώτηση 5/7

Τι ακριβώς κάνουν οι μέθοδοι `.seek()` και `.tell()` ;

Χειρίζονται τον *δείκτη ενός αρχείου* ο οποίος δείχνει πάντα την *διεύθυνση του byte* απ' όπου θα ξεκινήσει η επόμενη ενέργεια (*διάβασμα / γράψιμο*) στο αρχείο

- `.seek(k, w)` : μετακινεί τον δείκτη στη θέση που προσδιορίζεται από το `k`

Η επόμενη ενέργεια επάνω στο αρχείο (*εγγραφή ή ανάγνωση*) ξεκινάει από τη νέα αυτή θέση του δείκτη

Το προαιρετικό `w` καθορίζει αν το `k` είναι απόλυτο (`k` θέσεις από την *αρχή* –ή το *τέλος*- του αρχείου) ή σχετικό (`k` θέσεις από την *παρούσα θέση* του δείκτη)

- `.tell()` : επιστρέφει την παρούσα θέση του δείκτη (*στο άνοιγμα, είναι 0*)

Το `k` αναφέρεται πάντα σε bytes ανεξάρτητα αν το αρχείο είναι *κειμένου* ή *δυναμικό*

Ερώτηση 6/7

Ποιά είναι η *πλήρης μορφή* της μεθόδου `seek()` για δυαδικά αρχεία;

`seek(n, whence)` (*whence* : από πού;)

n : ακέραια τιμή που θα χρησιμοποιηθεί για να υπολογιστεί η *νέα θέση του δείκτη*

whence: (*σημείο αναφοράς*) προαιρετική ακέραια τιμή που υποδηλώνει ποια θέση του αρχείου θα αποτελέσει *το σημείο αναφοράς για τον υπολογισμό της νέας θέσης*. Μπορεί να πάρει τις τιμές 0, 1 ή 2 ως εξής:

- **0** Μετακίνηση n bytes από αρχή του αρχείου (*απόλυτη μετακίνηση*)
Η τιμή του **n** πρέπει να είναι πάντα *θετική* ή **0**
- **1** : Μετακίνηση n bytes από την τρέχουσα θέση του δείκτη (*σχετική μετακίνηση*)
Η τιμή του **n** μπορεί να είναι *θετική* ή *αρνητική*
- **2** : Μετακίνηση n bytes από το τέλος του αρχείου (*σχετική από το τέλος*)
Η τιμή του **n** μπορεί να είναι *θετική* ή *αρνητική*. Αν είναι θετική, τότε το αρχείο *επιμηκύνεται* κατά **n** θέσεις, οι οποίες γεμίζουν με τον χαρακτήρα που έχει κωδικό ASCII **0** (*συμβολίζεται \0*)
Προσοχή: ο χαρακτήρας \0 [κωδικός ASCII 0] είναι διαφορετικός από το ψηφίο "0" [κωδικός ASCII 48]

Όταν η παράμετρος **whence** απουσιάζει, τότε θεωρείται πως έχει την τιμή **0** (*default*)

Bonus Ερώτηση 7/7

Πως μπορώ να δημιουργήσω ένα νέο φάκελο και να γράψω μέσα σε αυτόν ένα αρχείο;

```
import os

# Define the directory name
dir_name = "new_directory"

# Create the directory
if not os.path.exists(dir_name):
    os.mkdir(dir_name)

# Change the current working directory to the new directory
os.chdir(dir_name)

# Create a text file in this new directory
with open('test.txt', 'w') as f:
    f.write("Hello!")
```

Το πακέτο *os* (*operating system*) περιέχει μεθόδους αλληλεπίδρασης με το λειτουργικό σύστημα, όπως δημιουργία και, διαχείριση καταλόγων, διαχείριση διεργασιών κ.λπ.