

Βασικές ασκήσεις, Διαφορές χρήσης
αρχείων, Παραδείγματα, Φυλλάδιο
Ερωτήσεων

Αρχεία

στην Python 2

ΧΡΗΣΤΟΣ ΜΑΒΟΓΛΟΥ

Περιεχόμενα

Ενότητα 1 – Βασικές Ασκήσεις για ανοίγματα αρχείων (open)	2
Ενότητα 2 – Ασκήσεις Ανάγνωσης με read() και readline()	9
Ενότητα 3 – Ασκήσεις με for line in file.....	11
Ενότητα 4 – Πολύ αναλυτικές Ασκήσεις για seek().....	13
Αναλυτικά οι διαφορές: w, r, a, r+	18
4 ασκήσεις, μία για κάθε mode (r, w, a, r+) - Εκφωνήσεις	27
Λύσεις των Ασκήσεων	31
Διαγώνισμα 10 Ερωτήσεων – File Modes (r, w, a, r+).....	36

Ενότητα 1 – Βασικές Ασκήσεις για ανοίγματα αρχείων (open)

Άσκηση 1.1 – Δημιουργία αρχείου χωρίς χρήση .close()

Στόχος: Ο μαθητής να δει ότι χωρίς close() το αρχείο μπορεί να μην γραφτεί σωστά.

Βήματα Μεθοδολογίας

1. Ανοίγουμε αρχείο με "w".
2. Γράφουμε κάτι.
3. **Δεν** καλούμε .close().
4. Ελέγχουμε το αρχείο.

Παράδειγμα:

```
f = open("test_no_close.txt", "w")  
f.write("Hello World WITHOUT close()")  
# f.close() # το αφήνουμε επίτηδες σχολιασμένο
```

Αναμενόμενη παρατήρηση:

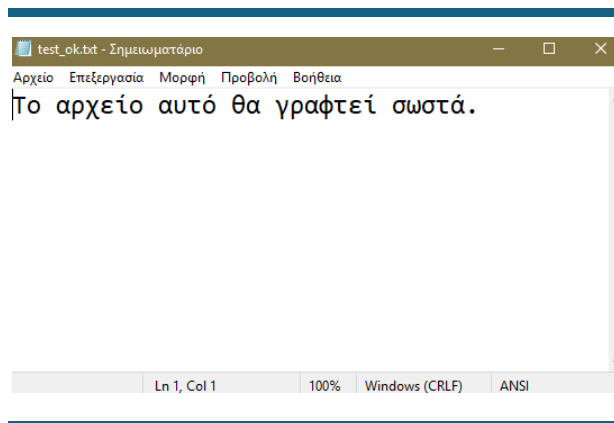
- Μπορεί το αρχείο να **μην αποθηκεύσει** το κείμενο ή να λείπουν δεδομένα.
 - Στη Python 2 οι buffers μπορεί να μην αδειάσουν.
 - Η σωστή πρακτική είναι *πάντα* close() ή χρήση with.
-

Άσκηση 1.2 – Δημιουργία αρχείου με σωστή χρήση .close()

```
f = open("test_ok.txt", "w")  
f.write("Το αρχείο αυτό θα γραφτεί σωστά.")  
f.close()
```

Συμπέρασμα:

- Η χρήση close() εξασφαλίζει ότι ο buffer αδειάζει → τα δεδομένα αποθηκεύονται.



Το αρχείο δημιουργείται.
Αν υπάρχει, τα
προηγούμενα
περιεχόμενα χάνονται

Άσκηση 1.3 – Χρήση open με mode "w" (δημιουργία / overwrite)

Στόχος: Το "w" δημιουργεί το αρχείο και διαγράφει παλιό περιεχόμενο, αν υπάρχει.

Παράδειγμα:

```
f = open("students.txt", "w")  
f.write("Maria\n")  
f.write("Nikos\n")  
f.close()
```

Παρατηρήσεις:

1. Αν το αρχείο υπάρχει, σβήνονται τα προηγούμενα περιεχόμενα, άρα το "w" θέλει προσοχή.
 2. Στη μέθοδο write το όρισμα πρέπει να είναι υποχρεωτικά string.
 3. Το: \n, αλλάζει σειρά στην εγγραφή. Χρησιμοποιείται συχνά, ώστε οι εγγραφές μας να βρίσκονται σε τάξη. Πρέπει να βρίσκεται μέσα σε εισαγωγικά σαν ένα string.
-

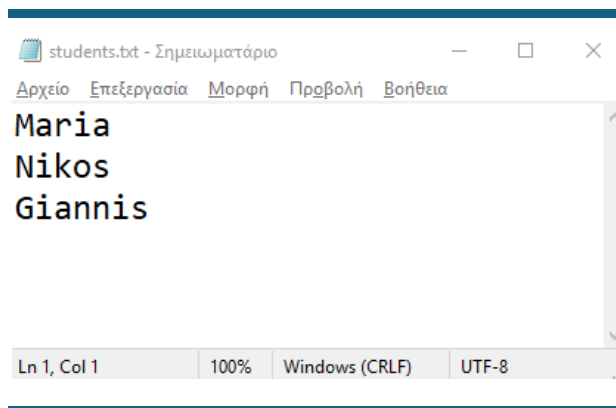
Άσκηση 1.4 – Χρήση open με mode "a" (append)

Στόχος: Να δει ο μαθητής ότι δεν διαγράφεται το αρχείο, απλώς προστίθενται γραμμές στο τέλος.

Παράδειγμα:

```
f = open("students.txt", "a")  
f.write("Giannis\n")  
f.close()
```

Τώρα το αρχείο έχει όσες γραμμές υπήρχαν + μία ακόμα.



Το αρχείο, αν υπάρχει,
προστίθενται οι νέες
εγγραφές στο τέλος του.

Αν δεν υπάρχει δημιουργείται.

Άσκηση 1.5 – Χρήση open με mode "r" (read only)

Στόχος: Να δει ότι για να διαβάσει πρέπει το αρχείο να υπάρχει.

Παράδειγμα:

```
f = open("students.txt", "r")
x = f.read()
print x
print 'Type of x: ', type(x)
f.close()
```

Αν το αρχείο δεν υπάρχει:

Η Python θα δώσει μήνυμα:

IOError: [Errno 2] No such file or directory

```
>>>
Maria
Nikos
Giannis

Type of x: <type 'str'>
>>>
```

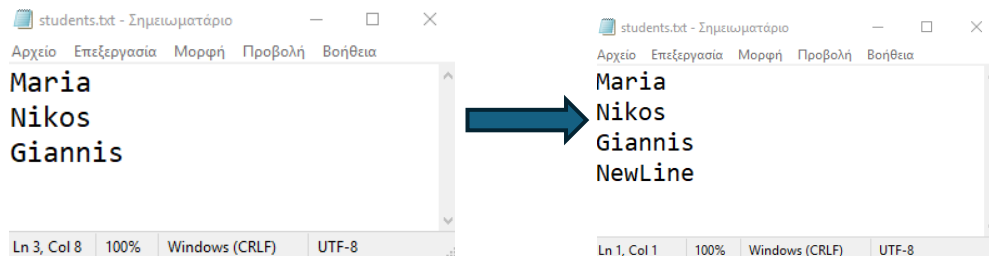
Τα περιεχόμενα του αρχείου
διαβάζονται όλα σε ένα string.

Άσκηση 1.6 – Χρήση open με mode "r+" (read AND write)

Στόχος: Μπορούμε να διαβάσουμε και να γράψουμε χωρίς να διαγραφούν τα περιεχόμενα του αρχείου.

Παράδειγμα 1:

```
f = open("students.txt", "r+")
f.read()           # διαβάζουμε
f.seek(0,2)        # πάμε στο τέλος
f.write("NewLine\n") # γράφουμε
f.close()
```

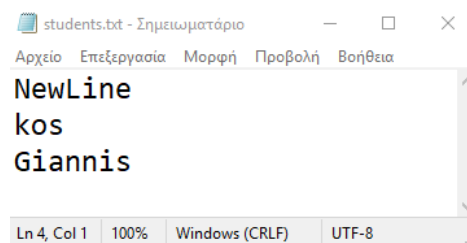
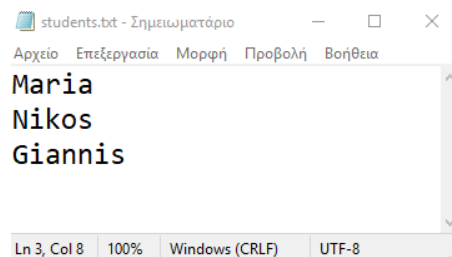


Προσοχή:

1. Το αρχείο **ΠΡΕΠΕΙ** να υπάρχει. Αν το αρχείο δεν υπάρχει, εμφανίζει το μήνυμα: `IOError: [Errno 2] No such file or directory: 'students.txt'`
2. Όταν χρησιμοποιούμε το mode "r+" και προηγείται ανάγνωση με `read()`, η μέθοδος **`.seek()` είναι υποχρεωτική**. Η εντολή: `f.seek(0, 2)` μεταφέρει τον δείκτη αρχείου (cursor) στο τέλος του αρχείου.
3. Αν αλλάξουμε τις παραμέτρους της `f.seek(0, 0)`, τότε η εγγραφή γίνεται στην αρχή του αρχείου, γράφοντας πάνω στους χαρακτήρες που υπάρχουν.

Παράδειγμα 2:

```
f = open("students.txt", "r+")  
f.read()           # διαβάζουμε  
f.seek(0,0)        # πάμε στην αρχή  
f.write("NewLine\n") # γράφουμε  
f.close()
```



Ενότητα 2 – Ασκήσεις Ανάγνωσης με read() και readline()

Για τις επόμενες ασκήσεις, δίνουμε έτοιμο αρχείο:

Περιεχόμενο αρχείου: poem.txt

Roses are red

Violets are blue

Python is fun

And so are you

Άσκηση 2.1 – Ανάγνωση ολόκληρου αρχείου με read()

```
f = open("poem.txt", "r")
text = f.read()
print text
f.close()
```

Μεθοδολογία:

- Το read() διαβάζει **όλο το αρχείο ως ένα string**.
- Οι αλλαγές γραμμής (\n) εμφανίζονται όπως είναι.

Άσκηση 2.2 – Ανάγνωση γραμμής-γραμμής με readline()

```
f = open("poem.txt", "r")

line = f.readline()
while line != "":
    print "Διαβάστηκε:", line
    line = f.readline()

f.close()
```

Μεθοδολογία:

- Το readline() επιστρέφει μια γραμμή.
 - Όταν φτάσει στο τέλος → επιστρέφει "".
-

Ενότητα 3 – Ασκήσεις με `for line in file`

Άσκηση 3.1 – Ανάγνωση όλων των γραμμών με `for`

```
f = open("poem.txt", "r")  
for line in f:  
    print ">", line  
f.close()
```

Άσκηση 3.2 – Αντιγραφή αρχείου με for

```
src = open("poem.txt", "r")
dst = open("copy_poem.txt", "w")

for line in src:
    dst.write(line)

src.close()
dst.close()
```

Ενότητα 4 – Πολύ αναλυτικές Ασκήσεις για seek()

Θα χρησιμοποιήσουμε νέο αρχείο **abc.txt**:

Περιεχόμενο αρχείου abc.txt

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Η seek(pos, mode) λειτουργεί ως εξής:

- **mode = 0** → από αρχή
- **mode = 1** → από τρέχουσα θέση
- **mode = 2** → από τέλος

Άσκηση 4.1 – Μετακίνηση στην αρχή του αρχείου

```
f = open("abc.txt", "r")
f.seek(0, 0)           # αρχή
print f.read(5)        # ABCDE
f.close()
```

Άσκηση 4.2 – Μετακίνηση 10 χαρακτήρες από την αρχή

```
f = open("abc.txt", "r")  
f.seek(10, 0)  
print f.read(5)          # KLMNO  
f.close()
```

Άσκηση 4.3 – Μετακίνηση σε σχέση με την τρέχουσα θέση (mode = 1)

```
f = open("abc.txt", "r")

print f.read(5)          # ABCDE
f.seek(3, 1)             # πήδα 3 χαρακτήρες από εδώ
→ πάμε στο H
print f.read(5)          # HIJKL

f.close()
```

Άσκηση 4.4 – Μετακίνηση σε σχέση με το τέλος (mode = 2)

```
f = open("abc.txt", "r")
```

```
f.seek(-5, 2)          # 5 πριν το τέλος
```

```
print f.read()         # VWXYZ
```

```
f.close()
```

Άσκηση 4.5 – Συνδυαστική άσκηση με πολλαπλές seek

```
f = open("abc.txt", "r")

f.seek(0, 0)
print f.read(3)          # ABC

f.seek(5, 0)
print f.read(2)          # FG

f.seek(-3, 2)
print f.read()           # XYZ

f.seek(2, 0)
print f.read(1)          # C

f.close()
```

1. Mode "r" – Read only (ΜΟΝΟ ανάγνωση)

1.1. Χαρακτηριστικά

- Σημαίνει **read**.
- Το αρχείο **πρέπει να υπάρχει**.
- Επιτρέπεται **μόνο ανάγνωση**, ΟΧΙ εγγραφή.
- Ο δείκτης αρχείου πάει **στην αρχή** (θέση 0).
- Αν το αρχείο **δεν υπάρχει** → **σφάλμα** (IOError).

1.2. Τι μπορούμε να κάνουμε

- `f.read()`
- `f.readline()`
- `for line in f:`
- Δεν επιτρέπεται `f.write()` (θα βγάλει σφάλμα).

1.3. Παράδειγμα

students.txt πρέπει να υπάρχει ήδη

```
f = open("students.txt", "r") # μόνο ανάγνωση
```

```
text = f.read()          # διαβάζει όλο το αρχείο
```

```
print text
```

```
f.close()
```

1.4. Τυπική παγίδα

Αν γράψεις:

```
f = open("students.txt", "r")
```

```
f.write("Katì\n")
```

θα πάρεις σφάλμα γιατί το "r" δεν επιτρέπει εγγραφή.

2. Mode "w" – Write (ΕΓΓΡΑΦΗ από την αρχή, ΜΗΔΕΝΙΣΜΟΣ)

2.1. Χαρακτηριστικά

- Σημαίνει **write**.
- Αν το αρχείο **δεν υπάρχει**, το **δημιουργεί**.
- Αν το αρχείο **υπάρχει**, το περιεχόμενο **διαγράφεται εντελώς** (truncate σε 0).
- Επιτρέπεται **μόνο εγγραφή**, όχι ανάγνωση (στην Python 2, πρακτικά δεν θα χρησιμοποιείς read σε "w").
- Ο δείκτης μπαίνει **στην αρχή** ενός **άδειου** αρχείου.

2.2. Τι μπορούμε να κάνουμε

- `f.write(...)`
- Όχι `f.read()` (δεν έχει νόημα, το αρχείο θεωρείται άδειο και mode είναι για write).

2.3. Παράδειγμα

Αν `students.txt` υπήρχε, τώρα θα σβηστεί το περιεχόμενο

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

```
f.write("Maria\n")
```

```
f.write("Nikos\n")
```

```
f.close()
```

Μετά την εκτέλεση, το αρχείο **περιέχει μόνο:**

Maria

Nikos

Οτιδήποτε υπήρχε πριν, χάθηκε.

2.4. Τυπικές παγίδες

1. Χρησιμοποιείς "w" αντί για "a" και χάνεις παλιά δεδομένα.
 2. Προσπαθείς να **διαβάσεις** αμέσως μετά το άνοιγμα με "w":
 3. `f = open("students.txt", "w")`
 4. `print f.read()` # δε δουλεύει όπως περιμένεις
-

3. Mode "a" – Append (ΠΡΟΣΘΗΚΗ στο τέλος)

3.1. Χαρακτηριστικά

- Σημαίνει **append** (πρόσθεσε στο τέλος).
- Αν το αρχείο **δεν υπάρχει**, το **δημιουργεί**.
- Αν το αρχείο υπάρχει, **ΔΕΝ διαγράφει** το περιεχόμενο.
- Ο δείκτης πάει **πάντα στο τέλος** του αρχείου.
- Γράφεις **μόνο στο τέλος** — δεν πειράζεις ό,τι υπάρχει ήδη.
- Είναι **κατάλληλο** για logs, λίστες καταγραφής κ.λπ.

3.2. Τι μπορούμε να κάνουμε

- `f.write(...)` → προσθέτει στο τέλος.
- Θεωρητικά μπορείς να κάνεις και `read`, αλλά στην πράξη με "a" το χρησιμοποιούμε **μόνο για γράψιμο**.

3.3. Παράδειγμα

Προσθήκη στο τέλος χωρίς να χαθούν τα παλιά

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

```
f.write("Eleni\n")
```

```
f.write("Giannis\n")
```

```
f.close()
```

Αν πριν είχες:

Maria

Nikos

Μετά θα έχεις:

Maria

Nikos

Eleni

Giannis

3.4. Τυπικές παγίδες

- Αν ανοίγεις πάντα με "a", το αρχείο **μεγαλώνει συνεχώς** και ίσως έχεις διπλές εγγραφές.
 - Αν νομίζεις ότι με "a" μπορείς να τροποποιείς "στη μέση" του αρχείου – **όχι**.
Με "a" γράφεις μόνο στο **τέλος**.
-

4. Mode "r+" – Read & Write (Ανάγνωση ΚΑΙ εγγραφή)

4.1. Χαρακτηριστικά

- Συνδυάζει "r" και "w" **χωρίς** διαγραφή.
- Το αρχείο **πρέπει να υπάρχει ήδη** (όπως "r").
- Επιτρέπεται **ανάγνωση και εγγραφή**.
- Ο δείκτης ξεκινάει **στην αρχή** του αρχείου.
- Δεν γίνεται **truncate** (δεν μηδενίζεται το αρχείο).
- Όταν γράφεις, από εκεί που είναι ο δείκτης και μετά, τα bytes **αντικαθίστανται**, δεν «σπρώχνονται».

4.2. Τι μπορούμε να κάνουμε

- `f.read()`, `f.readline()`, `for line in f` → πλήρης ανάγνωση.
- `f.write(...)` → γράφουμε όπου βρίσκεται ο δείκτης.
- `f.seek(pos, mode)` → μετακινούμε τον δείκτη όπου θέλουμε.

4.3. Παράδειγμα: Ανάγνωση + προσθήκη στο τέλος

```
f = open("students.txt", "r+")
```

```
print "=== Παλιά περιεχόμενα ==="
```

```
print f.read()
```

```
f.seek(0, 2)      # πάμε στο τέλος (mode 2 = από το τέλος)
```

```
f.write("NewLine\n") # προσθέτουμε νέα γραμμή στο τέλος
```

```
f.close()
```

4.4. Παράδειγμα: Αντικατάσταση στη μέση

```
f = open("students.txt", "r+")
```

```
f.seek(0, 0)    # αρχή
```

```
f.write("XXX")  # γράφει πάνω στα πρώτα 3 bytes
```

```
f.close()
```

Αν το αρχείο ξεκινούσε π.χ. με:

Maria

Θα γίνει:

XXXia

(γιατί αντικαθίσταται το "Mar" με "XXX").

4.5. Τυπικές παγίδες

1. Νομίζεις ότι "r+" κάνει «insert» χαρακτήρες.
→ Όχι, κάνει **overwrite** (αντικατάσταση).
 2. Ανοίγεις "r+" σε αρχείο που δεν υπάρχει.
→ σφάλμα, όπως με "r".
 3. Ξεχνάς ότι, μετά από ένα read(), ο δείκτης είναι στο τέλος. Αν γράψεις **χωρίς seek**, γράφεις στο τέλος.
-

5. Σύνοψη – Πίνακας σύγκρισης

	Πρέπει να Mode υπάρχει αρχείο;	Σβήνει τα παλιά;	Διαβάζει;	Γράφει;	Πού είναι ο δείκτης στην αρχή;
r	Ναι	Όχι	Ναι	Όχι	Στην αρχή
w	Όχι (το φτιάχνει)	Ναι	Όχι	Ναι	Στην αρχή άδειου αρχείου
a	Όχι (το φτιάχνει)	Όχι	Όχι (πρακτικά)	Ναι	Στο τέλος (γράφει μόνο στο τέλος)
r+	Ναι	Όχι	Ναι	Ναι	Στην αρχή

4 ασκήσεις, μία για κάθε mode (r, w, a, r+) - Εκφωνήσεις

Άσκηση 1 – Mode "r" (read)

Σκοπός: Να διαβάσει το αρχείο χωρίς να το τροποποιεί.

Δίνεται το αρχείο students.txt:

Maria

Nikos

Eleni

Να γραφεί πρόγραμμα σε Python 2 το οποίο:

1. Ανοίγει το αρχείο **students.txt** σε mode "r".
2. Διαβάσει και εμφανίζει **όλο το περιεχόμενο** του αρχείου.
3. Στη συνέχεια διαβάσει το αρχείο **γραμμή-γραμμή με readline()**.
4. Δεν επιτρέπεται καμία εγγραφή στο αρχείο.

Τι πρέπει να προσέξετε:

- Το αρχείο πρέπει **να υπάρχει**.
 - Δεν χρησιμοποιούμε write().
-

Άσκηση 2 – Mode "w" (write – overwrite)

Σκοπός: Κατανόηση ότι το "w" διαγράφει ό,τι υπήρχε πριν.

Να γραφεί πρόγραμμα το οποίο:

1. Ανοίγει αρχείο **grades.txt** με mode "w".
2. Γράφει 3 βαθμούς, έναν σε κάθε γραμμή.
3. Κλείνει το αρχείο.
4. Ξανατρέχεις το πρόγραμμα και παρατηρείς τι συμβαίνει με το παλιό περιεχόμενο.

Τι πρέπει να προσέξετε:

- Το "w" **μηδενίζει** το αρχείο κάθε φορά.
 - Η πρώτη εκτέλεση και η δεύτερη εκτέλεση **παράγουν διαφορετικό αποτέλεσμα** στο αρχείο.
-

Άσκηση 3 – Mode "a" (append – προσθήκη στο τέλος)

Σκοπός: Πρόσθεση εγγραφών χωρίς να σβήνονται τα παλιά.

Δίνεται το αρχείο log.txt:

Start Program

User logged in

Να γραφεί πρόγραμμα που:

1. Ανοίγει το **log.txt** με mode "a".
2. Προσθέτει στο τέλος τη γραμμή:

New session started

3. Κλείνει το αρχείο.

Τρέχεις το πρόγραμμα και ελέγχεις πώς μεγαλώνει το αρχείο.

Τι πρέπει να προσέξετε:

- Με "a" ο δείκτης **είναι πάντα στο τέλος**.
 - Δεν πρέπει να χρησιμοποιηθεί για ανάγνωση/τροποποίηση στη μέση του αρχείου.
-

Άσκηση 4 – Mode "r+" (read & write)

Σκοπός: Συνδυασμός ανάγνωσης και εγγραφής μέσα στο αρχείο, χωρίς διαγραφή.

Δίνεται το αρχείο poem.txt:

Roses are red

Violets are blue

Python is fun

Να γραφεί πρόγραμμα που:

1. Ανοίγει το αρχείο **poem.txt** με mode "r+".
2. Διαβάζει και εμφανίζει **το περιεχόμενο** με read().
3. Με seek(-4, 2) μετακινείται 4 χαρακτήρες πριν το τέλος του αρχείου.
4. Γράφει πάνω στα τελευταία γράμματα τη λέξη:
NEW
5. Κλείνει το αρχείο.
6. Ξανανοίγει το αρχείο και εμφανίζει το νέο περιεχόμενο.

Τι πρέπει να προσέξετε:

- Το "r+" **δεν κάνει insert**, κάνει **overwrite**.
 - Το αρχείο πρέπει **να υπάρχει ήδη**.
 - Το offset (ο πρώτος δείκτης) με αρνητικό αριθμό χρησιμοποιεί μέτρηση από το τέλος.
-

Λύσεις των Ασκήσεων

Λύση Άσκησης 1 – Mode "r" (read μόνο)

Εκφώνηση (υπενθύμιση):

Να ανοίξεις το students.txt σε "r", να εμφανίσεις όλο το περιεχόμενο με read(), και μετά γραμμή-γραμμή με readline().

Άσκηση 1 – Λύση με mode "r"

Πρώτος τρόπος: read() – διαβάζει όλο το αρχείο

f = open("students.txt", "r") # το αρχείο πρέπει να υπάρχει

contents = f.read()

print "=== Περιεχόμενο με read() ==="

print contents

f.close()

Δεύτερος τρόπος: readline() – διαβάζει γραμμή-γραμμή

f = open("students.txt", "r")

print "=== Περιεχόμενο με readline() ==="

line = f.readline() # διαβάζει την πρώτη γραμμή

while line != "": # όσο δεν φτάσαμε στο τέλος

print line

line = f.readline() # επόμενη γραμμή

f.close()

Λύση Άσκησης 2 – Mode "w" (write – overwrite)

Εκφώνηση (υπενθύμιση):

Να ανοίξεις grades.txt με "w", να γράψεις 3 βαθμούς (έναν ανά γραμμή), να κλείσεις το αρχείο. Αν τρέξεις ξανά το πρόγραμμα, να παρατηρήσεις ότι τα παλιά δεδομένα χάνονται.

Άσκηση 2 – Λύση με mode "w"

Κάθε φορά που εκτελείται αυτό το πρόγραμμα,

το grades.txt ΜΗΔΕΝΙΖΕΤΑΙ και ξαναγράφεται από την αρχή.

```
f = open("grades.txt", "w") # δημιουργεί νέο ή σβήνει το παλιό
```

```
f.write("18\n")
```

```
f.write("15\n")
```

```
f.write("20\n")
```

```
f.close()
```

```
print "Το αρχείο grades.txt γράφτηκε ξανά με 3 βαθμούς."
```

(Η παρατήρηση για το ότι σβήνει τα παλιά περιεχόμενα γίνεται από τον/την μαθητή/τρια ανοίγοντας το αρχείο πριν/μετά.)

Λύση Άσκησης 3 – Mode "a" (append – προσθήκη στο τέλος)

Εκφώνηση (υπενθύμιση):

Δίνεται log.txt. Να ανοίξεις με "a", να προσθέσεις τη γραμμή "New session started" στο τέλος.

Άσκηση 3 – Λύση με mode "a"

```
f = open("log.txt", "a") # άνοιγμα για προσθήκη στο τέλος
```

```
f.write("New session started\n")
```

```
f.close()
```

```
print "Προστέθηκε νέα γραμμή στο τέλος του log.txt."
```

Αν τρέξεις το πρόγραμμα πολλές φορές, η γραμμή θα εμφανίζεται **πολλαπλές φορές** στο τέλος του αρχείου (λογική log).

Λύση Άσκησης 4 – Mode "r+" (read & write)

Εκφώνηση (υπενθύμιση):

Δίνεται poem.txt. Να το ανοίξεις με "r+", να διαβάσεις και να εμφανίσεις το περιεχόμενο, να κάνεις seek(-4, 2) και να γράψεις "NEW", μετά να κλείσεις και να ξαναδιαβάσεις το νέο περιεχόμενο.

Άσκηση 4 – Λύση με mode "r+"

1. Άνοιγμα για ανάγνωση + εγγραφή, το αρχείο πρέπει να υπάρχει

```
f = open("poem.txt", "r+")
```

```
print "=== Αρχικό περιεχόμενο poem.txt ==="
```

```
old_text = f.read()
```

```
print old_text
```

2. Μετακίνηση 4 χαρακτήρες πριν το τέλος του αρχείου

```
f.seek(-4, 2) # offset = -4, mode = 2 (από το τέλος)
```

3. Εγγραφή "NEW" πάνω στα τελευταία 4 bytes

```
f.write("NEW") # προσοχή: κάνουμε overwrite, όχι insert
```

```
f.close()
```

4. Ξανά άνοιγμα και εμφάνιση του νέου περιεχομένου

```
f = open("poem.txt", "r")
```

```
print "=== ΝΕΟ περιεχόμενο poem.txt ==="
```

```
print f.read()
```

```
f.close()
```

Σημεία για τον/την μαθητή/τρια:

- Το `seek(-4, 2)` μετράει 4 χαρακτήρες από το τέλος προς τα πίσω.
 - Το "NEW" θα αντικαταστήσει **3** από αυτούς τους χαρακτήρες (γιατί έχει 3 γράμματα).
 - Δεν «ανοίγει χώρο», απλώς **πατάει πάνω** στα προηγούμενα bytes.
-

Διαγώνισμα 10 Ερωτήσεων – File Modes (r, w, a, r+)

Ερώτηση 1

Με ποιο mode ανοίγουμε ένα αρχείο αποκλειστικά για **ανάγνωση**, χωρίς δυνατότητα αλλαγής περιεχομένου;

- α) "w"
 - β) "a"
 - γ) "r"
 - δ) "r+"
-

Ερώτηση 2

Ποιο mode **διαγράφει** το προηγούμενο περιεχόμενο του αρχείου και ξεκινά νέο αρχείο από την αρχή;

- α) "a"
 - β) "w"
 - γ) "r"
 - δ) "r+"
-

Ερώτηση 3

Ποιο mode χρησιμοποιούμε όταν θέλουμε να **προσθέσουμε** νέες γραμμές *χωρίς* να αλλάξουν οι ήδη υπάρχουσες;

- α) "w"
 - β) "a"
 - γ) "r+"
 - δ) "r"
-

Ερώτηση 4

Ποιο mode επιτρέπει **ταυτόχρονα** ανάγνωση και εγγραφή;

- α) "r"
 - β) "a"
 - γ) "w"
 - δ) "r+"
-

Ερώτηση 5

Όταν ανοίγουμε αρχείο με mode "w" και γράψουμε μέσα κείμενο, το νέο περιεχόμενο:

- α) προστίθεται στο τέλος
 - β) μπαίνει στην αρχή και σβήνει ό,τι υπήρχε
 - γ) εισάγεται στη μέση του αρχείου
 - δ) αντικαθιστά μόνο τα πρώτα bytes χωρίς να διαγράψει τα υπόλοιπα
-

Ερώτηση 6

Με ποιο mode ο δείκτης βρίσκεται **πάντα στο τέλος** του αρχείου, ανεξάρτητα από προηγούμενες εντολές;

- α) "a"
 - β) "w"
 - γ) "r"
 - δ) "r+"
-

Ερώτηση 7

Ποιο mode επιτρέπει εγγραφή, αλλά **χωρίς** να σβηστούν τα προηγούμενα δεδομένα;

- α) "r"
 - β) "w"
 - γ) "a"
 - δ) "r+"
-

Ερώτηση 8

Σε ποιο mode η εντολή write() αντικαθιστά τα δεδομένα **από τη θέση του δείκτη και μετά**, χωρίς εισαγωγή νέων δεδομένων στη μέση;

- α) "r"
 - β) "a"
 - γ) "r+"
 - δ) "w"
-

Ερώτηση 9

Στο mode "r+", μετά την εκτέλεση της read(), ο δείκτης πηγαίνει:

- α) στην αρχή του αρχείου
 - β) στο τέλος του αρχείου
 - γ) στη μέση του αρχείου
 - δ) σε τυχαία θέση
-

Ερώτηση 10

Ποιο mode είναι καταλληλότερο για να διαβάσουμε πρώτα το αρχείο και μετά να προσθέσουμε κάτι στο τέλος του;

α) "r" + seek(0, 0)

β) "r+" + seek(0, 2)

γ) "w"

δ) "a" και μετά read()
