

Βασικές Ασκήσεις από τα Εργαστήρια της PYTHON (εβδομάδα 9)

► Τα προγράμματα που κάναμε στα προηγούμενα εργαστήρια διάβαζαν και έγραφαν στο standard input, standard output και standard error. Για να γράψουμε σε αρχείο και να διαβάσουμε από αρχείο πρέπει πρώτα να ανοίξουμε το αρχείο. Στο τέλος του προγράμματος πρέπει να το κλείσουμε.

Σύνταξη

```
file object = open(file_name [, access_mode][, buffering])
```

Παράμετροι:

- **file_name:** Το όνομα του αρχείου είναι string.
- **access_mode:** Ο τρόπος με τον οποίο επεξεργαζόμαστε το αρχείο.
- **Buffering:** 0 χωρίς buffering, 1 line buffering. Εάν δώσετε τιμή μεγαλύτερη του 1, τότε η τιμή αυτή καθορίζει το μέγεθος του buffer που χρησιμοποιείται για την πρόσβαση στο αρχείο.

Modes

Περιγραφή

r	Ανάγνωση μόνο. Ο file pointer τοποθετείται στην αρχή του αρχείου.
rb	Ανάγνωση binary αρχείου. Ο file pointer τοποθετείται στην αρχή του αρχείου.
r+	Ανάγνωση και γράψιμο. Ο file pointer τοποθετείται στην αρχή του αρχείου.
rb+	Ανάγνωση και γράψιμο σε binary format. Ο file pointer τοποθετείται στην αρχή του αρχείου.
w	Ανοίγει ένα αρχείο για γράψιμο μόνο. Καταστρέφει το αρχείο εάν υπάρχει ήδη. Εάν δεν υπάρχει το δημιουργεί.
wb	Ανοίγει ένα αρχείο για γράψιμο μόνο σε binary format. Καταστρέφει το αρχείο εάν υπάρχει ήδη. Εάν δεν υπάρχει το δημιουργεί.
w+	Ανάγνωση και γράψιμο. Καταστρέφει το αρχείο εάν υπάρχει ήδη. Εάν δεν υπάρχει το δημιουργεί.
wb+	Ανάγνωση και γράψιμο σε binary format. Καταστρέφει το αρχείο εάν υπάρχει ήδη. Εάν δεν υπάρχει το δημιουργεί.
a	Appending ή προσθήκη σε αρχείο. Ο file pointer τοποθετείται στο τέλος του αρχείου εάν το αρχείο υπάρχει. Εάν δεν υπάρχει το δημιουργεί.
ab	Appending ή προσθήκη σε αρχείο σε binary format. Ο file pointer τοποθετείται στο τέλος του αρχείου εάν το αρχείο υπάρχει. Εάν δεν υπάρχει το δημιουργεί.
a+	Appending ή προσθήκη σε αρχείο και διάβασμα. Ο file pointer τοποθετείται στο τέλος του αρχείου εάν το αρχείο υπάρχει. Το αρχείο ανοίγει σε append mode. Εάν δεν υπάρχει το δημιουργεί και το ανοίγει για γράψιμο και διάβασμα.
ab+	Appending ή προσθήκη σε αρχείο και διάβασμα σε binary format. Ο file pointer τοποθετείται στο τέλος του αρχείου εάν το αρχείο υπάρχει. Το αρχείο ανοίγει σε append mode. Εάν δεν υπάρχει το δημιουργεί και το ανοίγει για γράψιμο και διάβασμα.

► Το παρακάτω πρόγραμμα ανοίγει το αρχείο workfile για γράψιμο και γράφει:

```
f = open('workfile', 'w')
for i in range(0,10):
    f.write('line {0}\n'.format(i))

f.close()
```

Το αρχείο workfile αφού τρέξει το πρόγραμμα

```
line 0
line 1
line 2
line 3
line 4
line 5
line 6
line 7
line 8
line 9
```

► Το επόμενο πρόγραμμα διαβάζει το αρχείο workfile:

```
f = open('workfile', 'r')
for line in f:
    print(line)

f.close()
```

► Το αρχείο names.data:

```
Nikos
Maria
John
Stathis
```

► Αυτό το πρόγραμμα διαβάζει τα περιεχόμενα του αρχείου names.data και τυπώνει το μήκος κάθε ονόματος και το όνομα χωρίς το '\n' που το ακολουθεί:

```
f=open('names.data', 'r')  
  
for line in f:  
    n=len(line)  
    print(n,line[:n-1])  
  
f.close()
```

► Το πρόγραμμα διαβάζει ένα-ένα τα bytes του αρχείου και τα τυπώνει:

```
f = open("names.data", "r")  
  
byte = f.read(1)  
while byte:  
    print(byte)  
    byte = f.read(1)  
  
f.close()
```

Δίνει τα παρακάτω αποτελέσματα όταν τρέξει:

```
>>> ===== RESTART =====  
>>>  
N  
i  
k  
o  
s  
  
M  
a  
r  
i  
a  
  
...
```

Δοκιμάστε να αντικαταστήσετε το r με br:

```
f = open("names.data", "br")
```

► Το πρόγραμμα χρησιμοποιεί την δομή **try – except – catch** για έλεγχο επιτυχούς λειτουργίας των εντολών διαχείρισης αρχείων:

```
try:
    data = open('its.txt', "w")
    print("It's...", file=data)
except IOError as err:
    print('File error: ' + str(err))
finally:
    if 'data' in locals():
        data.close()
```

► Η χρήση του **with** ακυρώνει την ανάγκη χρήσης **finally** και αυτόματα καλεί την **close()**:

```
try:
    with open('its.txt', "w") as data:
        print("It's...", file=data)
except IOError as err:
    print('File error: ' + str(err))
```