

פרק 8 קבצים

תוכן העניינים :

פרק 8 : עבודה עם קבצים

2	8.1 מבוא
3	8.2 פתיחה של קובץ <code>open()</code>
3	8.3 התחביר לפתיחת קובץ
5	8.4 סגירת קובץ <code>close()</code>
5	8.5 טיפול בקובץ
6	8.6 קריאה מקובץ
7	8.7 כתיבה לקובץ <code>write()</code>
9	8.8 הוספה לקובץ – <code>append()</code>
11	8.8 תרגילים בקבצים

בפרק 8 של תוכנית הלימודים להנדסאים במקצוע שפת פייתון מופיע:

פרק 8 : עבודה עם קבצים

יעדים

הכרות עם מערכת הקבצים במחשב, מחיצות directories, קבצים files ומסלול path. שימוש בקבצים לקלט פלט לשמירה והעברת מידע וכזיכרון לא נדיף.

תכנים

1. מבוא לקבצים

2. ההבדל בין קבצי טקסט (Text Files) לקבצים בינאריים (Binary Files)

3. עבודה עם זרמי / שטפי מידע – Stream

4. מניפולציות על קבצים :

- open
- read
- close

5. התמודדות עם סוגי טעויות (חריגים – Exception) נפוצות :

- FileNotFoundError
- FileIsADirectoryException
- FileNotADirectoryException

סיכום שעות ההוראה : 12 שעות עיוניות ו- 4 שעות התנסותיות. סה"כ 14 שעות.

פרק 8 : עבודה עם קבצים

8.1 מבוא

טיפול בקבצים הוא חלק חשוב בכל עבודה עם מחשב ויישום אינטרנט. ל- Python מספר פונקציות ליצירה, קריאה, עדכון ומחיקה של קבצים.

8.1 א סוגי קבצים

ישנם 2 סוגי קבצים ששפת פייתון מטפלת בהם:

א. קובץ טקסט – Text File - שבו מאחסנים תווים. לדוגמה: myfile.txt. בקובץ כזה יש תווי סיום שורה '\n' – EOL שהוא קיצור של End Of Line בסיום כל שורה בקובץ.

ב. קובץ בינארי – Binary File - משמשים לאחסון נתונים בינאריים כמו תמונות, קבצי וידאו, קבצי שמע, קבצי מכונה -exe - וכו'.

8.1 ב מיקום של קובץ - Path – נתיב/מסלול

ה Path - נתיב/מסלול מגדיר את מיקום הקובץ או הספרייה במחשב. ישנן 2 דרכים לציין את ה Path :

א. נתיב מוחלט – Absolute Path – שמתחיל תמיד עם תיקיית הבסיס root folder – לדוגמה : c:\.

ב. נתיב יחסי - Relative Path – שהוא יחסי לספריית העבודה הנוכחית של התוכנית שאנחנו רושמים.

הנתיב המוחלט כולל את כל רשימת הספריות המלאה הדרושה למיקום הקובץ.

לדוגמה : c:\users\arik\marks.txt הוא נתיב מוחלט המציין את מיקום קובץ הטקסט marks (ציונים). לאחר שם הקובץ, החלק עם נקודה(.) נקרא סיומת הקובץ, וזה אומר לנו את סוג הקובץ שבדוגמה כאן הוא קובץ טקסט.

8.1.1 פתיחת ספריית עבודה

לפני עבודה עם קבצים המתוארים בסעיפים הבאים מומלץ ליצור ספרייה בכונן C הנקראת pythonTextFiles ובה נכתוב/נקרא קבצים. הנתיב יהיה : c:\pythonTextFiles\כולל סיומת\שם הקובץ

8.1.2 שלבים בטיפול בקובץ

- כמו בשפות אחרות, גם בפיתון יש 3 שלבים בטיפול בקובץ :
- פתיחת הקובץ.
 - קריאה או כתיבה או הוספה לקובץ.
 - סגירה של הקובץ.

8.2 פתיחה של קובץ open()

- הפונקציה לעבודה עם קבצים ב-Python היא הפונקציה **open()**.
- הפונקציה מקבלת 2 פרמטרים :
- את שם הקובץ שבו רוצים לטפל
 - את אופן העבודה עם הקובץ (קריאה או כתיבה או הוספה) .

קיימות ארבע שיטות שונות (מצבים) לפתיחת קובץ:

- "r"** – קריאה read . פתיחת הקובץ לקריאה בלבד **מתחילתו** של הקובץ. אם הקובץ איננו קיים נקבל הודעת שגיאה.
 - "a"** – הוספה append – פתיחת הקובץ להוספה **בסוף** הקובץ. אם הקובץ איננו קיים אז נוצר קובץ חדש בשם שנתנו לקובץ.
 - "w"** – כתיבה write – פתיחת קובץ **לכתיבה בהתחלת** הקובץ. אם הקובץ איננו קיים אז נוצר קובץ חדש בשם שנתנו לקובץ.
 - "x"** – create – יצירה של הקובץ עם השם שנתנו. אם הקובץ כבר קיים נקבל הודעת שגיאה.
- כתוספת, ניתן לציין האם הקובץ יטופל כקובץ טקסט או קובץ בינארי.
- "t"** – text – ברירת המחדל הקובץ הוא קובץ טקסט. (אם אחרי r או w לא רושמים כלום).
 - "b"** – binary – הקובץ הוא קובץ בינארי (כמו קבצי תמונה, ריצה וכו').

- ניתן להוסיף + אחרי אופן העבודה והדבר מוסיף אופן עבודה נוסף. לדוגמה : rt+ אומר לקרוא קובץ טקסט וגם לכתוב אליו. דוגמה נוספת : a+ אומר שפותרים את הקובץ להוספה וגם לקריאה.

8.3 התחביר לפתיחת קובץ

התחביר לפתיחת קובץ :

```
File_object = open(r"File_Name", "Access_Mode")
```

```
"אופן העבודה", "שם הקובץ") = open(r
```

```
fp = open(r"doogma.txt", "rt") : לדוגמה
```

fp הוא "אובייקט קובץ" בעזרתו נפעיל מתודות העובדות עם קבצים.

התו - r - הממוקם לפני שם הקובץ משמש כדי למנוע התייחסות לתווים במחרוזת שם הקובץ כאל תו מיוחד. לדוגמה, אם יש ספרייה עם שם כמו temp \ בכתובת הקובץ, ולא רוצים שהיא תטופל כמו תו הטאב (tab) \t ואז נקבל שגיאה של כתובת לא חוקית. ה-r הופך את המחרוזת לגולמית (raw), כלומר, היא אומרת שהמחרוזת היא ללא תווים מיוחדים. ניתן להתעלם מה-r אם הקובץ נמצא באותה ספרייה והכתובת אינה ממוקמת בספרייה עם \t בהתחלה.

שם הקובץ – את שם הקובץ רושמים בין גרשיים. אם הקובץ לא נמצא בספרייה הנוכחית יש לרשום את ה path .

אופן העבודה וסוג הקובץ – האם קריאה או/ו כתיבה או הוספה ומהו סוג הקובץ, האם קובץ טקסט (t) או קובץ בינארי (b).

דוגמה 1: פתיחת קובץ טקסט לקריאה כאשר הקובץ נמצא בספרייה הנוכחית

```
f = open("myfile.txt", "rt")
```

פתיחת קובץ בשם myfile.txt . פתיחת הקובץ היא לצורך קריאה של טקסט .

בדוגמה הזו ניתן גם לרשום f = open("myfile.txt") מהסיבה ש-r לקריאה ו-t עבור טקסט הם ערכי ברירת המחל. הערה: כמובן שיש לוודא שהקובץ קיים אחרת נקבל שגיאה.

דוגמה 2: פתיחת קובץ טקסט לכתיבה כאשר הקובץ נמצא בספרייה הנוכחית

```
f = open("myfile1.txt", "wt")
```

הערה: יש לשים לב שהכתיבה מתבצעת מתחילת הקובץ ומה שהיה רשום בקובץ נמחק. כמו כן אם הקובץ איננו קיים ייווצר קובץ חדש עם השם myfile1.txt .

דוגמה 3: פתיחת קובץ עם ציון נתיב מוחלט

```
fp = open(r"c:\pythonTtextFiles\sample1.txt", "r")
```

בדוגמה כזו נקבל הודעת שגיאה אם לא נשים את ה r בגלל שקיימת ספרייה \textFiles

דוגמה 4: פתח 2 קבצים הנקראים demo1.txt ו demo2.txt . הקבצים נמצאים בספרייה שבה אנחנו רושמים את התוכנית.

```
fp1 = open(demo1.txt", "r")
```

```
fp2 = open(demo1.txt", "r")
```

פתחנו כאן 2 אובייקטים של קובץ . לאחד קוראים fp1 ולשני קוראים fp2 .

כמובן שאם אחד או שני קבצי הטקסט לא קיימים נקבל הודאת שגיאה.

דוגמה 5: נפתח קובץ הנמצא במיקום אחר מזה שהתוכנית נמצאת בו. לשם כך ניצור קובץ טקסט בעזרת פנקס הרשימות או הכתבן ונקרא לו demo1 (הקובץ יקבל סיומת txt לבד). נשמור את הקובץ בכונן C בספרייה C:\pythonTextFiles. אם נרצה לראות מה רשום בקובץ נרשום:

```
fp1 = open("C:\pythonTextFiles\demo1.txt","r")
fp2 = open("C:\pythonTextFiles\demo2.txt", "r")
```

8.3.1 ההצהרה with open()

ניתן לפתוח קובץ עם ההצהרה with open(). הדבר גורם לסגור את הקובץ מחוץ לבלוק באופן אוטומטי. התחביר הכולל את with הוא מקובל ונוח ומרבים להשתמש בו. לדוגמה:

```
with open ("c:\pythonTextFiles\demo1.txt","r") as p1:
    print( p1.readlines())
```

ההדפסה שנקבל:

```
['Hello world\n', 'Have a nice day\n', 'Hope every thing will be OK for you']
```

בסיום לולאת with open הקובץ ייסגר ללא צורך לרשום p1.close()

אם ננסה לבצע קריאה נוספת מהקובץ בעזרת הפקודה print(p1.readlines()) נקבל את הודעת השגיאה:

Traceback (most recent call last):

```
File "<pyshell#7>", line 1, in <module>
    print( p1.readlines())
```

ValueError: I/O operation on closed file.

8.4 סגירת קובץ close()

מומלץ תמיד לסגור את הקובץ כשמסיימים לעבוד איתו. הסגירה מתבצעת עם הפונקציה close(). ועם אובייקט הקובץ איתו פתחנו את הקובץ:

התחביר: **close().** אובייקט הקובץ

אם פתחנו את הקובץ עם השורה: fp1 = open(demo1.txt", "r") אז כאשר נסגור את הקובץ נרשום:

```
fp1.close()
```

8.5 טיפול בקובץ

נניח שיש לנו קובץ בשם demo1.txt שנמצא בנתיב C:\pythonTextFiles\demo1.txt. בקובץ רשום:

Hello world

Have a nice day

Hope every thing will be OK for you

D:\My Documents\אריק\python\targilim\demo1.txt : אותו הקובץ נמצא גם בספרייה :

נפתח את הקובץ עם הפקודה : `fp1 = open(r"D:\My Documents\אריק\python\targilim\demo1.txt", "r")`
שבה מגדירים אובייקט קובץ `fp1` ופותחים את הקובץ לקריאת טקסט .
הערה : כדאי לשים לב שהשתמשנו פה בתו `r` לפני ציון הנתיב כי בקובץ יש גם ספרייה בעברית ואם לא נעשה זאת נקבל הודעת שגיאה.
השימוש ב `r` יהיה גם אם בנתיב יש ספרייה שמתחילה בתו `t` כי אז יהיה נדמה שמדובר בתו `t` שהוא העברת הסמן לשדה ההדפסה הבא (בדומה למקש `tab`) .
הטיפול בקובץ יכול להיות קריאה , כתיבה , קריאה וכתיבה, הוספה (כתיבה בסוף הקובץ), הוספה וקריאה.

8.6 קריאה מקובץ

ישנן 3 פונקציות המטפלות בקריאה מקובץ . `read()` , `readline()` ו `readlines()` .

8.6 א קריאה של הטקסט שבקובץ `read()`

הפונקציה קוראת את כל הטקסט שבקובץ . אם נרשום :

```
fp1 = open("C:\pythonTextFiles\demo1.txt","r")
print(fp1.read())
```

נקבל :

```
Hello world
Have a nice day
Hope every thing will be OK for you
```

8.6 א.1 קריאה של חלקים מתוך הקובץ

ברירת המחדל של המתודה `read()` היא להחזיר את כל הטקסט שבקובץ. ניתן לציין כמה תווים רוצים שהפונקציה תחזיר .
דוגמה : החזרה של 5 התווים הראשונים בקובץ

```
fp1 = open("C:\pythonTextFiles\demo1.txt","r")
print(fp1.read(5))
```

ההדפסה שנקבל : Hello

8.6 ב קריאה של שורה `readline()`

ניתן להחזיר שורה אחת מהקובץ על ידי שימוש במתודה `readline()`
דוגמה 1:

```
fp1 = open("C:\pythonTextFiles\", "r")
print(fp1.readline())
```

ההדפסה שנקבל:

Hello world

אם נרשום בהמשך :

```
print(fp1.readline())
```

Have a nice day

נקבל את השורה הבאה :

דוגמה 2: רשום תוכנית כדי להדפיס את כל הטקסט שבקובץ בלולאה :

```
fp1 = open("C:\pythonTextFiles\", "r")
```

```
for x in fp1:
```

```
    print(x)
```

ההדפסה שנקבל :

Hello world

Have a nice day

Hope every thing will be OK for you

8.6.g קריאה של שורות `readlines()`

הפונקציה קוראת את כל השורות ומחזירה כל שורה במחרוזת בצורת רשימה.

התחביר : `File_object.readlines()` לדוגמה : `fp1.readlines()`

הערה: `'\n'` מטופל כתו מיוחד של שני בתים

דוגמה:

```
>>> fp1 = open("C:\pythonTextFiles\", "r")
```

```
>>> print(fp1.readlines())
```

ההדפסה שנקבל :

```
['Hello world\n', 'Have a nice day\n', 'Hope every thing will be OK for you']
```

8.7 כתיבה לקובץ `write()`

קיימות שתי פונקציות לכתוב לקובץ.

1. `write()` : הוספת מחרוזת `str1` בשורה אחת בקובץ הטקסט. `File_object.write(str1)`

2. `writelines()` : עבור רשימה של מחרוזות. כל מחרוזת נוספת לקובץ הטקסט. הפונקציה משמשת להוספת

מחרוזות מרובות בו-זמנית.

עבור הרשימה $L = [str1, str2, str3]$ נכתוב `File_object.writelines(L)`
הערה : כדאי לזכור שאם קיים הקובץ אז בזמן הכתיבה יימחק כל מה שהיה לפני כן בקובץ. אם לא קיים ייוצר הקובץ.

דוגמה 1 : נבצע את השלבים הבאים : נפתח קובץ `demo2.txt` לכתיבה, נכתוב טקסט לקובץ, נסגור את הקובץ. נפתח אותו לקריאה ונדפיס את מה שרשום בקובץ ובסיום התוכנית נסגור את הקובץ.

```
>>> p1 = open(r"C:\pythonTextFiles\demo2.txt", "w")
>>> p1.write("Hello yood dalet \n")
18 # כמות התווים שנכתבה לקובץ
>>> p1.close()
```

```
>>> p1 = open(r"C:\pythonTextFiles\demo2.txt", "r")
>>> print(p1.read())
Hello yood dalet
>>> p1.close()
```

דוגמה 2 : נבצע את השלבים הבאים : נפתח קובץ `demo3.txt` לכתיבה ונכתוב אליו רשימה של מחרוזות. נסגור את הקובץ, נפתח את הקובץ לקריאה ונדפיס את הכתוב בו. בסוף התוכנית נסגור את הקובץ.

```
>>> list1 = ["kita yood dalet \n", "This is new year \n", "we learn in Tel Aviv \n"]
>>> p1 = open(r"C:\pythonTextFiles\demo3.txt", "w")
>>> p1.writelines(list1)
>>> p1.close()
```

```
>>> p1 = open(r"C:\pythonTextFiles\demo3.txt", "r")
>>> print(p1.read())
kita yood dalet
This is new year
we learn in Tel Aviv
>>> p1.close()
```

דוגמה 3 : העתקה של בלוק:

```
with open("C:\pythonTextFiles\demo3.txt", "r") as rp :
    with open("C:\pythonTextFiles\demo4.txt", "w") as wp :
        for line in rp:
            wp.write(line)
```


כדאי לשים לב. הלולאה הראשונה קוראת את הקובץ בעזרת אובייקט הקובץ rp והלולאה שמתחתיה כותבת את הקובץ בעזרת אובייקט הקובץ wp .
נקבל הדפסה של כמות התווים שנכתבת לקובץ עבור כל שורה.

17

18

22

29

הערה : אין צורך לסגור את הקבצים. הם נסגרים לבד בגלל המשפט with open .

דוגמה 4 : העתקת קובץ תמונה . במקרה כזה נוסיף את אופן הפעולה rb לקריאה ו wb לכתיבה. התו b מראה שמדובר בקובץ בינארי.

```
with open("C:\pythonTextFiles\picture1.jpg", "rb") as rp :  
    with open("C:\pythonTextFiles\picture2.jpg", "wb") as wp :  
        for line in rp:  
            wp.write(line)
```

דוגמה 5 : העתקת קובץ לפי בלוקים

כאשר יש להעתיק קובץ גדול מאוד כדאי לחלק את הקובץ הגדול לבלוקים/חתיכות בגודל קטן יותר כדי לא להעמיס את זיכרון המטמון של המחשב (cash memory) .

```
blockSize = 2048  
with open("C:\pythonTextFiles\picture3.jpg", "rb") as rp :  
    with open("C:\pythonTextFiles\picture4.jpg", "wb") as wp :  
        block = rp.read(blockSize) # קריאה של 2048 תווים  
        while len(block)>0: # כל עוד קיבלנו בלוק עם כמות תווים גדולה מ 0 ?  
            wp.write(block) # כתיבת הבלוק לקובץ  
            block = rp.read(blockSize)
```

8.8 הוספה לקובץ - Appending to a file

כאשר רוצים להוסיף טקסט בסוף קובץ משתמשים באופן העבודה "a" שהוא קיצור של append .
דוגמה : הוסף לקובץ demo3.txt את המחרוזת " we added this string to file "

```
>>> p = open(r"C:\pythonTextFiles\demo3.txt", "a")  
>>> p.write("we added this string to file\n")
```

29

כמות התווים שנכתבה לקובץ

```
>>> p.close()
```

```
>>> p = open(r"C:\pythonTextFiles\demo3.txt", "r") :
```

```
>>> print(p.read())
```

ההדפסה שנקבל:

kita yood dalet

This is new year

we learn in Tel Aviv

we added this string to file

8.9 תרגילים בקבצים

שאלה 1

רשום את תוכנית שתבצע את הפעולות הבאות : א. תפתח קובץ בשם tar1.dat ב. תקלוט מהמשתמש מחרוזות (עד 255 תווים) ג. תכתוב לקובץ את המחרוזות ד. תסגור את הקובץ.

שאלה 2

רשום תוכנית שתספור כמה תווים נשמרו בקובץ (כמה תווים הקיש המשתמש).

שאלה 3

כתוב תוכנית שתקלוט מהמשתמש נתונים בלולאה עד שהוא רושם exit . הקלט יודפס לקובץ טקסט כאשר כל נתון בשורה חדשה. בסיום התוכנית הדפס כמה שורות המשתמש הכניס.

שאלה 4

רשום תוכנית שתבדוק ותדפיס כמה פעמים מופיע כל תו בקובץ שבשאלה הראשונה.

שאלה 5

קלוט מהמשתמש 5 מחרוזות אל רשימה. את הרשימה של 5 המחרוזות שמור בקובץ בשם tar2.dat . בין מחרוזות אחת לאחרת הכנס את התו שורה חדשה (\n - צרוף של CR/LF).

שאלה 6

רשום תוכנית שתספור כמה שורות יש בקובץ שבשאלה הקודמת.

שאלה 7

רשום תוכנית שתעתיק את כל הנתונים שבקובץ שבשאלה 5 tar2.dat אל קובץ נוסף בשם tar3.dat.

שאלה 8

רשום תוכנית שתעתיק את הקובץ tar1.dat אל קובץ אחר tar4.dat אבל בסדר הפוך, כלומר התו האחרון של הקובץ יהיה ראשון ולהיפך. (רמז – העבר את הקובץ למחרוזת והעתק את המחרוזת הפוך)

שאלה 9

רשום תוכנית שתציג מידע לגבי הקובץ tar2.dat. המידע יהיה : כמה תווים יש עם אותיות גדולות (מ A ועד Z), כמה תווים עם אותיות קטנות (מ a ועד z), כמה תווים הם ספרות (1 עד 9) וכמה תווים "לבנים" (תווים שהם לא אחד מהתווים הקודמים).

שאלה 10

רשום תוכנית שתקלוט מהמשתמש תו ותדפיס למסך כמה פעמים מופיע התו הנקלט בקובץ tar2.dat . התוכנית תרוץ שוב בלולאה עד שהמשתמש יקיש על מקש escape .

שאלה 11

קלוט מהמשתמש רשימה בת 50 מספרים מטיפוס שלם והכנס אותם לקובץ tar5.dat . רשום קטע תוכנית למציאת והדפסת המספר הגדול והמספר הקטן שבקובץ וכמה פעמים כל אחד מהם מופיע.

שאלה 12 (שאלה ברמה גבוהה יותר)

כתוב תוכנית המקבלת מהמשתמש נתיב – path – ומציגה ברשימה את הקבצים בספרייה זו. לדוגמה אם המשתמש נתן את הנתיב : c:\pythonTextFiles יירשם :

1. demo1.txt
2. demo2.txt
3.

לאחר מכן תינתן למשתמש האופציה לבחור מספר והתוכנית תמחק את הקובץ.

שאלה 13 (שאלה ברמה גבוהה יותר)

כתוב תוכנית שמקבלת מהמשתמש נתיב לקובץ . יש להדפיס את הפרטים הבאים לגבי הקבצים בספרייה :

- שם הקובץ
- כמות התווים בקובץ
- כמות השורות בקובץ
- כמה פעמים מופיעה המילה "we" .

שאלה 14 (שאלה ברמה גבוהה יותר)

כתוב תוכנית שמקבלת מהמשתמש נתיב של קובץ והיא תצנזר מילים שהוגדרו לצינזור. את המילים שיש לצנזר יש לשמור בקובץ נפרד הנקרא censoredWords.txt . יש לשים כוכביות עבור הצנזור.