

# פייתון

שיעור 8: Files and Script Parameters

## פתיחת קובץ: הפונקציה open ▶

- מקבלת שם קובץ ו-mode של פתיחה
- r- read, w- write (לקבצי טקסט)
- r-rb (read binary), w- wb (לקבצים שכוללים מידע לא טקסטואלי)
- a - append (כתיבה מסוף הקובץ, אחרת יש דריסה)
- מחזירה מצביע ל-file
- לאובייקט file יש מתודות (פונקציות פנימיות- נלמד בהמשך)

```
>>> input_file = open(r'c:\cyber\dear pr  
>>> for line in input_file:  
    print line,
```

הקובץ לא  
נשמר  
בזיכרון

הפסיק מגדיר שלא לרדת  
שורה בסוף ה-print, כיוון  
שהקובץ כבר כולל את  
הוראות ירידת השורה

```
"Dear Prudence" / The Beatles
```

```
Dear Prudence, won't you come out to play?  
Dear Prudence, greet the brand new day  
The sun is up, the sky is blue  
It's beautiful and so are you  
Dear Prudence, won't you come out to play?
```



- ▶ המתודה read קוראת את הקובץ כולו ושומרת אותו בזיכרון
- מאד נוח
- פחות מתאים לקבצים עצומים

```
>>> input_file = open(r'c:\cyber\dear_prudence.txt', 'r')
>>> lyrics = input_file.read()
>>> print lyrics
"Dear Prudence" / The Beatles
```

```
Dear Prudence, won't you come out to play?
Dear Prudence, greet the brand new day
The sun is up, the sky is blue
It's beautiful and so are you
Dear Prudence, won't you come out to play?
```

כיוון שהקובץ כבר מכיל מידע, נפתח אותו ב-append

```
input_file = open(r'c:\python\dear_prudence.txt', 'a')  
input_file.write('Dear Prudence open up your eyes\n')
```

```
FILENAME = r'c:\python\dear_prudence.txt'
```

```
}def open_without_closing(filename):  
    fd1 = open(filename, 'a')  
}    fd1.write('Dear Prudence open up your eyes\n')  
  
}def main():  
    open_without_closing(FILENAME)  
    fd2 = open(FILENAME, 'r')  
    for line in fd2:  
}        print line,
```

# close, with

- ▶ לאחר שסיימנו את העבודה על קובץ, חשוב לסגור אותו
- ▶ `input_file.close()`
- ▶ ההוראה `with` גורמת לכך שפונקציית הסגירה של הקובץ מופעלת באופן אוטומטי
  - המתודה תופעל בסוף הבלוק שמתחיל ב-`with` - כך שניתן להשתמש בקובץ כל עוד אנחנו בבלוק
  - תכנות נכון- אינו משאיר קבצים פתוחים

```
with open(FILENAME, 'r') as input_file:  
    for line in input_file:  
        print line,
```

# תרגיל- מכונת שכפול

- ▶ צרו באמצעות סייר חלונות שני קבצי טקסט, אחד מהם ריק.
- ▶ כיתבו סקריפט שיעתיק אל הקובץ הריק את הקובץ השני.



# קבלת פרמטרים לתוכנית

- ▶ לעיתים נרצה להעביר לתוכנית פרמטרים טרם ההרצה
- ▶ נעדכן את `base.py`:

- `import sys` – מאפשר קריאה לפונקציות של חבילת תוכנה `module` בשם `sys`
- `-sys.argv` – שומר את הפרמטרים שהעברנו ב-`command line`
- עוד על `sys` ניתן לקרוא ב: [http://www.python-course.eu/sys\\_module.php](http://www.python-course.eu/sys_module.php)

```
import sys
```

```
def main():  
    print sys.argv
```

```
if __name__ == '__main__':  
    main()
```



# שימוש בפרמטר ספציפי

```
import sys
NAME = 1

def main():
    print 'Hello {}'.format(sys.argv[NAME])

if __name__ == '__main__':
    main()
```

Hello Shooki

Process finished with exit code 0

- ▶ לתוך `sys.argv` נשמרים הפרמטרים שהזין המשתמש
- ▶ נוצרת רשימה `list`
  - באיבר ה-0- שם התוכנית
  - באיבר ה-1- הפרמטר הראשון
  - וכו'
  - קריאה לאיבר הראשון ברשימה:  
`sys.argv[1]`

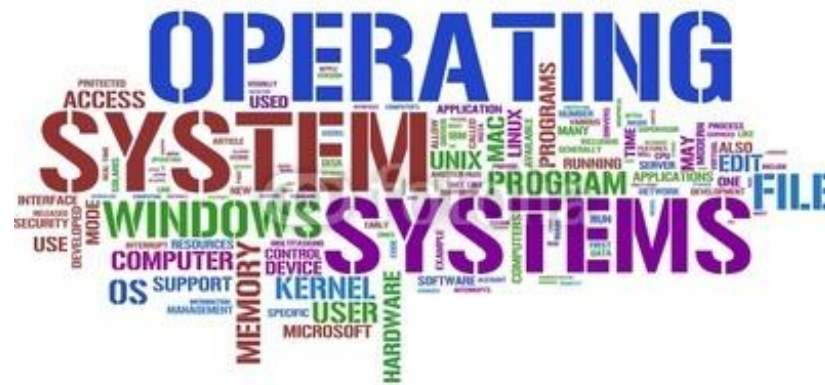
# תרגיל Printer

▶ כיתבו תוכנית שמקבלת כפרמטר שם של קובץ, ומדפיסה את התוכן שלו למסך.



# מודול שימושי: OS

- ▶ בתרגיל printer, מה יקרה אם הקובץ אינו קיים?
- ▶ OS - קיצור של Operating System
- ▶ המודול עוסק בפעולות של מערכת הפעלה כגון ניהול ספריות, הצגת קבצים בספרייה וכו'



```
import sys
import os
PATH = 1

def main():
    directory = sys.argv[PATH]
    print os.listdir(directory)

if __name__ == '__main__':
    main()
```

- ▶ דוגמה: קבלת path והצגת שמות הקבצים בספריה
  - path - נתיב של ספריה. הכוונה לשרשרת הספריות מה-root עד לספריה המבוקשת
  - תזכורת: ניתן לקבוע את הפרמטרים ב-pycharm דרך תפריט ה-edit configurations
  - סימן '.' (נקודה) מציין: הספריה בה נמצא סקריפט הפייתון שנמצא בהרצה

- ▶ המודול `os.path` כולל אוסף של מתודות שימושיות למציאת `path`, מעבר ל-`path` אחר וכו'.
  - ניתן למצוא מידע על ידי `dir` או בחיפוש ברשת



```
import sys
import os
PATH = 1

def main():
    directory = sys.argv[PATH]
    print os.listdir(directory)

if __name__ == '__main__':
    main()
```



- ▶ העתיקו והריצו את הקוד הבא, אך העבירו לסקריפט כפרמטר path שאינו קיים
  - מה קרה לתוכנית?
- ▶ כעת, פיתרו את הבעיה בעזרת מתודה של os, אשר מקבלת path ובודקת אם הוא קיים
- ▶ טיפ: חפשו מתודה בעזרת dir(os.path)

# תרגיל - Lazy Student

- ▶ קיבלתם כשיעורי בית קובץ עם תרגילי חשבון
- ▶ כל תרגיל הוא מהצורה מספר-רווח-פעולה-רווח-מספר

◦ לדוגמה:  $46 + 19$

◦ פעולה יכולה להיות  $+/-/*$  בלבד

- ▶ כיתבו סקריפט פייתון שמקבל קובץ תרגילים ושומר לקובץ את כל התרגילים פתורים

◦ לדוגמה  $46 + 19 = 65$

◦ הסקריפט יקבל כפרמטרים שמות של 2 קבצים - מקור ופתרון

- ▶ הניחו שיש שגיאות פורמט בחלק מהתרגילים - או בשם הקובץ - אסור לסקריפט לקרוס בשום אופן.

BILL GATES SAYS :

I WILL ALWAYS CHOOSE A LAZY PERSON  
TO DO A DIFFICULT JOB ...  
BECAUSE, HE WILL FIND AN EASY  
WAY TO DO IT.

