

map, filter, lambda

הפונקציה map בפייתון, משמשת כדרך להפעיל פונקציה כלשהיא שנרצה, על סידרת ערכים (עצם איטרטיבי) בשורה בודדת.

לדוגמה, נניח שיש לנו רשימת מספרים חיוביים, ועבור כל מספר עלינו ליצור מספר אקראי בין 0 למספר עצמו, ומספרים אקראים אלה יהוו רשימה חדשה.

הרשימה המקורית:

```
lst1 = [10, 19, 30, 7]
```

היינו יכולים לפתור זאת עם הקוד הבא:

```
import random
lst_rand = []
for num in lst1:
    lst_rand.append(random.randrange(num))
```

או שהיינו יכולים גם לכתוב:

```
import random
lst_rand = list(map(random.randrange, lst1))
```

כלומר הפונקציה map ממפה לנו פונקציה על סידרת ערכים, או מפעילה אותה על סדרת הערכים. הפונקציה ש map מקבלת יכולה גם להיכתב על ידנו, למשל:

```
def far_to_cel(far):
    cel = (far - 32) * 5/9
    return round(cel, 1)
```

```
def main():
    lst1 = [30, 75, 95, 104]
    print (list(map(far_to_cel, lst1)))
```

שימו לב ש map מקבלת את הפונקציה כפרמטר ולכן אין סוגריים לאחר שם הפונקציה.

מה עושה filter ? היא מסננת תוצאות של סידרת ערכים על פי פונקציה שנגדיר. הפונקציה שנגדיר חייבת להחזיר ערך בוליאני כדי ש filter תעבוד (True או False)

לדוגמה נגדיר פונקציה שמחזירה אמת אם מספר הוא חיובי ושקר אחרת:

```
def positive(x):  
    return x > 0
```

```
def main():  
    lst1 = [-10, 19, 21, -1, 7, 10, 0]  
    only_positives = list(filter(positive, lst1))
```

שימו לב שגם במקרה של map וגם של filter השתמשנו במילת המפתח list כדי לקבל חזרה תוצאה. כלומר שתיהן מחזירות עצם איטרטיבי. אם היינו רק מדפיסים את התוצאה הישירה של map או filter היינו מקבלים את המיקום בזיכרון שהעצם נוצר, למשל:

```
print (filter(positive, lst1))
```

```
<filter object at 0x103c7ffd0>
```

כעת נראה מהו ביטוי למדא

ביטוי למדא לפעמים ניקרא פונקציה אנונימית, כלומר פונקציה ללא שם. איך זה יכול להיות, שהרי פונקציה נועדה להיקרא? אז זהו שפונקציה אנונימית מוגדרת וניקראת באותה נשימה (חד פעמית).

בואו וניקח למשל פונקציה שמחזירה ריבוע של מספר ונהפוך אותה לביטוי למדא:

```
def square(num):  
    return num**2
```

```
square = lambda num: num**2
```

ביצענו השמה של ביטוי למדא לשם square אבל אמרנו שלמדא היא אנונימית. אם נשתמש בביטוי למדא בהקשר של map ו/או filter נראה איך ביטוי הלמדא הוא כמו פונקציה חסרת שם לשימוש חד פעמי:

```
lst1 = [4, 10, 12, 13]  
lst2 = list(map(lambda num: num**2, lst1))
```

```
==> [16, 100, 144, 169]
```

לסיכום אומר שביטוי למדא היא למעשה צורה של פונקציה פשוטה שנכתבת בשורה עם תחביר מיוחד. קודם מילת המפתח lambda (לא הוגים את ה b), לאחר מכן הפרמטר (ים), נקודותיים ומימין לנקודותיים זה מה שעושים לו return בפונקציה רגילה (הערך המוחזר).

נראה למשל ביטוי למדא עם יותר מפרמטר אחד:

```
lambda x,y: x+y
```

הקוד הבא:

```
k = lambda x,y : x+y  
print(k(10,20))
```

ידפיס את המספר 30

עוד אציין, בשלב זה ללא פירוט נוסף, שיש מקרים בפייתון, בהם חייבים להשתמש
בביטוי למדא במקום בפונקציה רגילה, אחרת הקוד לא יעבוד.