

- הכרות עם שפת התכנות פייתון (PYTHON)
- הכרות עם סביבת העבודה של PYTHON
- תחביר השפה:

- |              |               |
|--------------|---------------|
| ✓ אופרטורים  | ✓ משתנים      |
| בוליאנים     | ✓ סוגי משתנים |
| ✓ משפטי תנאי | ✓ השמה        |
| ✓ לולאות     | ✓ הדפסה       |
|              | ✓ אופרטורים   |
|              | ✓ תיעוד       |

הכל נמצא ברשת, צריך רק לדעת איפה לחפש:

## אתרים ללמוד עצמי

- The Algorithm Design Manual by Steven S Skiena (highly recommended)
- [Python Documentation](#)
- [Think Python: How to Think Like a Computer Scientist](#) by Allen B. Downey
- [Python Programming](#) WikiBook
- [The Python Tutorial](#) official tutorial

➤ תחביר השפה:

✓ פונקציות

✓ מבנה נתונים

✓ ספריות

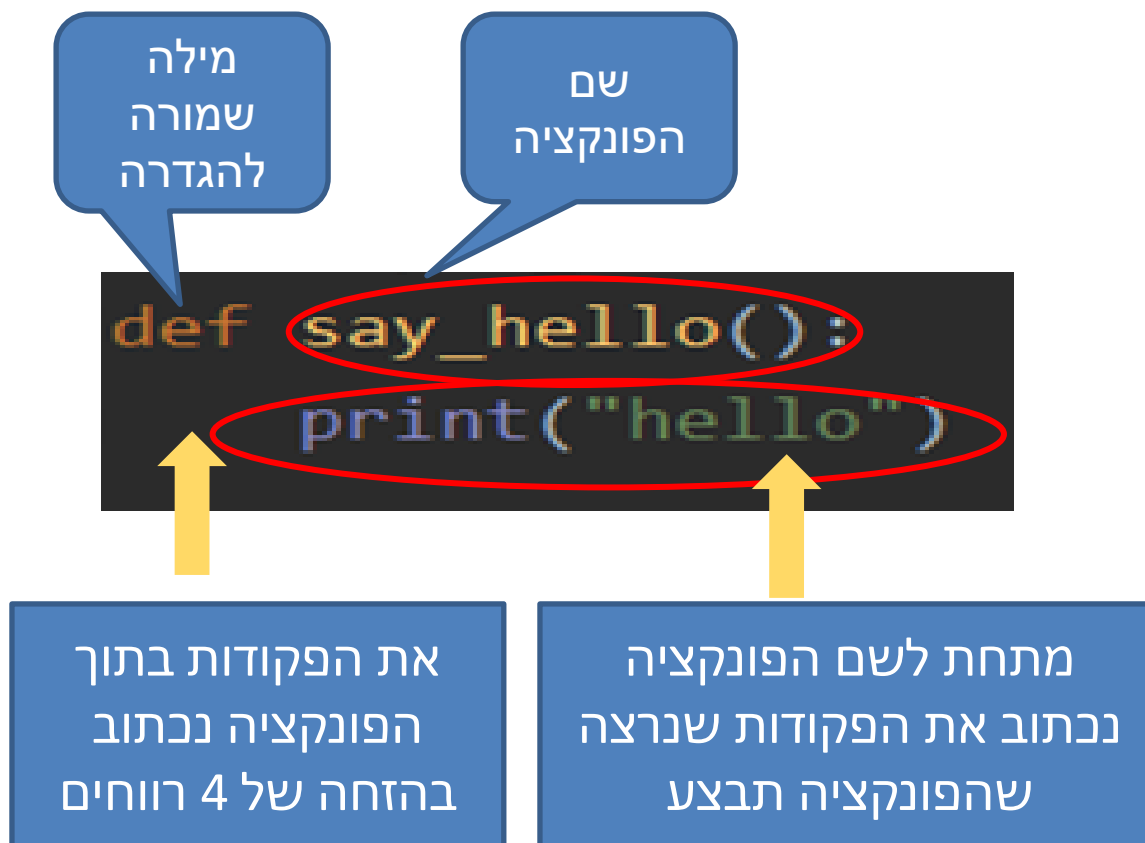
# פונקציות

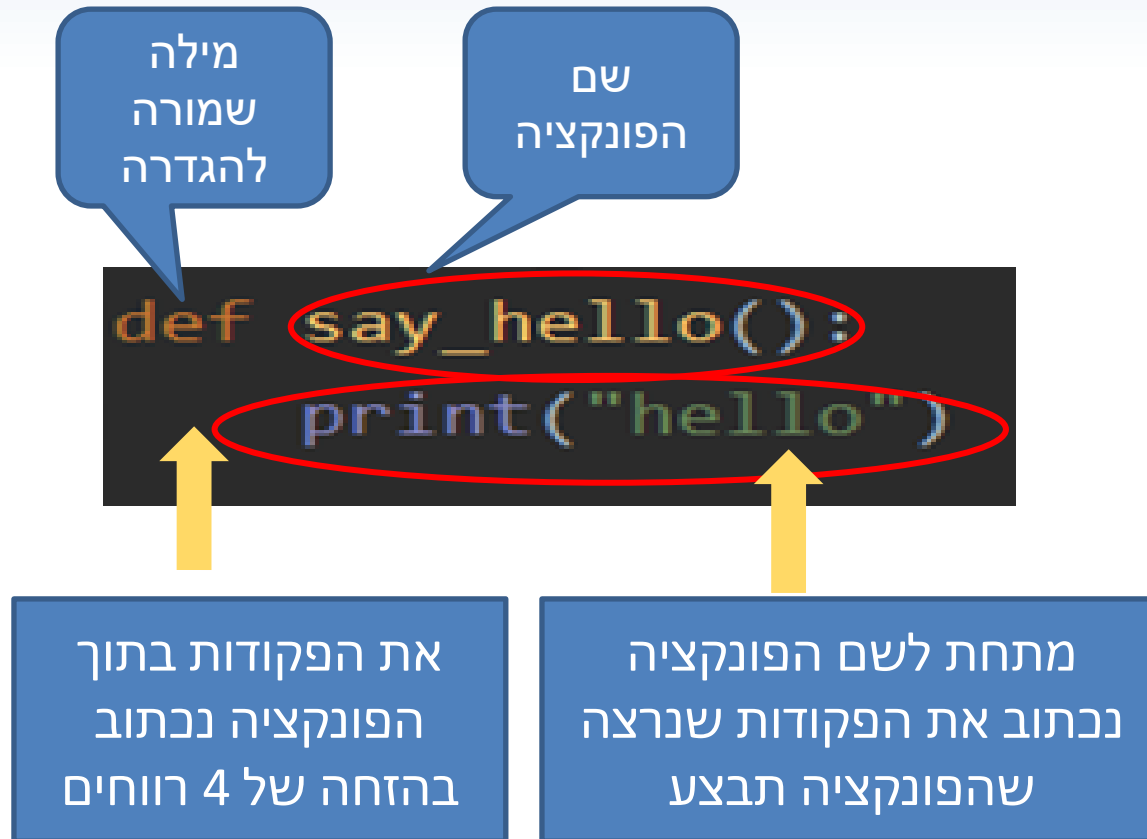
➤ **כתיבת פונקציה**

➤ **קריאה לפונקציה**

➤ **קלט ופלט**

- נשתמש בפונקציה כאשר נרצה לכתוב קטע קוד ו"לארוז" אותו כך שנוכל להפעיל אותו בלי לכתוב את אותו קוד שוב כל פעם.
- פונקציה מכילה מספר פקודות שמקובצות ביחד תחת שם אחד באופן שניתן לקרוא לה מאיפה שצריך, מתי שצריך.
- פונקציה יכולה לקבל פרמטרים כקלט ויכולה להחזיר פרמטרים בפלט
- נוכל לקרוא לפונקציה מתי שנרצה ובהתאם למטרות שלנו!





```
say_hello()
```

כדי שהפונקציה תרוץ אנחנו צריכים לקרוא לה בשמה באופן הבא:

```
def say_hello():  
    print("hello")  
  
say_hello()
```



הקוד המלא

הגדרת הפונקציה

קריאה לפונקציה

```
hello
```

אם נריץ את הקוד הפלט יהיה:



מה יודפס לאחר הרצת הקוד הבא?

```
def say_hello():  
    print("hello")  
  
say_hello()  
say_hello()
```

- א. hello  
hello  
hello
- ב. hello  
hello
- ג. hello hello
- ד. hello hello hello

מה יודפס לאחר הרצת הקוד הבא?

```
def say_hello():  
    print("hello")  
  
say_hello()  
say_hello()
```

ד.  
hello hello hello

ג.  
hello hello

ב.  
hello  
hello

א.  
hello  
hello  
hello

מה יודפס לאחר הרצת הקוד הבא?

```
for i in range(100):  
    say_hello()
```

ד.  
say hello

ג.  
hello

ב.  
99  
פעמים  
המילה  
hello

א.  
100  
פעמים  
המילה  
hello

מה יודפס לאחר הרצת הקוד הבא?

```
for i in range(100):  
    say_hello()
```

ד.  
say hello

ג.  
hello

ב.  
99  
פעמים  
המילה  
hello

א.  
100  
פעמים  
המילה  
hello

קלט

```
def say_hello(name):  
    print("hello", name)
```

הפונקציה מקבלת כקלט **שם** בפרמטר:

**name**

בתוך הפונקציה:

הדפסה של המחרוזת:

hello **ושם**

```
def say_hello(name):  
    print("hello", name)
```

```
say_hello("Shahar")  
say_hello("Kfir")  
say_hello("Ofek")
```

נוכל לקרוא לפונקציה שלנו עם שמות שונים  
וכך נקבל פלטים שונים:

```
def say_hello(name):  
    print("hello", name)
```

```
x = input("הכנס שם")  
say_hello(x)
```

נוכל לקרוא לפונקציה שלנו עם משתנה  
שערכו הוא שם שביקשנו ממשתמש:

כתוב פונקציה שמקבלת בקלט שני פרמטרים  
ומדפיסה את סכומם:

לדוגמא עבור קלט עם המספרים: 1, 4

יודפס:

The sum of 1 + 4 is: 5

פונקציה לא חייבת להדפיס פלט.  
היא גם יכולה להחזיר ערכים

```
def mul_numbers(num1, num2):  
    return num1 * num2
```



כדי להחזיר ערך  
מפונקציה נכתוב את  
הפקודה **return** ואחריה  
את הערך שנרצה להחזיר

פונקציה שמקבלת **בקלט** שני מספרים  
ומחזירה **כפלט** את המכפלה ביניהם



מה יוצג על המסך בהרצת הקוד הבא:

```
def mul_numbers(num1, num2):  
    return num1 * num2  
  
mul_numbers(10, 5)
```

ד.  
לא יוצג כלום

ג.  
50

ב.  
10,5

א.  
15

## תרגול 4 - פונקציות - פתרון

מה יוצג על המסך בהרצת הקוד הבא:

```
def mul_numbers(num1, num2):  
    return num1 * num2  
  
mul_numbers(10, 5)
```

א.  
15

ב.  
10,5

ג.  
50

ד.  
לא יוצג כלום

```
def mul_numbers(num1, num2):  
    return num1 * num2  
  
print(mul_numbers(10, 5))
```

כדי להציג את ערך ההחזרה של  
הפונקציה נצטרך להדפיס אותה

1

כתבו פונקציה המקבלת כפרמטרים שלושה מספרים ומחזירה את הממוצע שלהם

2

כתבו פונקציה המקבלת כפרמטר מספר ומדפיסה ריבוע מ-# שצלעו הוא גודל המספר.

לדוגמא: אם המספר שנשלח לפונקציה הוא 3 אז הפונקציה תדפיס –

```
###
###
###
```

3

כתבו פונקציה המקבלת כפרמטר מספר.

הפונקציה תחזיר True אם המספר הוא זוגי, אחרת תחזיר False.

4

כתבו פונקציה המקבלת בקלט מספר תלת ספרתי ומחזירה שלושה פרמטרים כפלט:

את ספרת האחדות של המספר, את ספרת העשרות של המספר ואת ספרת המאות של המספר

# מבנה נתונים

רשימה – LIST ➤

מילון - Dictionary ➤

**מבנה נתונים** הוא רכיב תוכנה המאפשר לנו:

✓ לשמור נתונים (ערכים),

✓ לבצע עליהם פעולות מסוגים שונים, כגון:

☐ הוספת נתון,

☐ מחיקת נתון,

☐ חיפוש נתון וכדומה

רשימה היא סדרה של ערכים  
הערכים יכולים להיות:

מספרים

מחרוזות

רשימות

ומבני נתונים אחרים נוספים.

כדי ליצור רשימה נבחר לה שם ונכניס בסוגריים מרובעים []  
את הערכים שנרצה לשמור בה מופרדים בפסיקים.

שם הרשימה  
(כמו במשתנים)

[הערכים של הרשימה,  
מופרדים בפסיקים]

```
my_list = [1, 2, 3, 4, 5]
```

ערכי הרשימה: מספרים מ 1 עד 5

ערכי הרשימה: מספר שלם,

מספר עשרוני, מחרוזת

ורשימה

```
my_list = [1, 3.5, "hello", [1, 4, "hi"]]
```

לכל איבר ברשימה יש מספר סידורי:

האיבר הראשון הוא האיבר ה-0

השני הוא ה-1

השלישי הוא ה-2 וכו'...

כדי לגשת לאיבר ברשימה נכתוב את שם הרשימה, בצמוד לשם סוגריים מרובעים, ובתוך הסוגריים את מספר האיבר שאנו רוצים לגשת אליו.



בהינתן הרשימה הבאה:

```
my_list = [1, 2, 3, 4, 5]
```

איך תציגו את האיבר הרביעי ברשימה?

ד.

Print(my\_list[0])

ג.

Print(my\_list[3])

ב.

Print(my\_list[4])

א.

Print(my\_list)

בהינתן הרשימה הבאה:

```
my_list = [1, 2, 3, 4, 5]
```

איך תציגו את האיבר הרביעי ברשימה?

ד.  
Print(my\_list[0])

ג.  
Print(my\_list[3])

ב.  
Print(my\_list[4])

א.  
Print(my\_list)

עדכון ערך לאיבר מסוים ברשימת `my_list` יתבצע כך :

```
my_list[1] = "two"
```

# פעולות מובנות על רשימות

שם	מה היא עושה?
len(A)	מחזירה את הערך של רשימה A
.sort()	ממיינת את הרשימה *A
.append(a)	מוסיף את האיבר a בסוף הרשימה
.remove(a)	מסיר את האיבר ה-a הראשון ברשימה
.pop()	מסיר את האיבר במקום האחרון ברשימה
.reverse()	הופך את סדר הרשימה
count(a)	סופר את מספר הפעמים שהאיבר a מופיע ברשימה
clear()	מוחק את כל האיברים מהרשימה
A[i]	גישה לאיבר במיקום ה-i

# הוספת ערך לרשימה - append

הוספת איבר לסוף רשימה קיימת באמצעות הפקודה

append

```
my_list.append("Bye")
```

ואם נדפיס את הרשימה כעת זה מה שיודפס לנו:

```
[1, 2, 3, 4, 5, 'Bye']
```

## ניתן לעבור בלולאת for על ערכי הרשימה

הדפסת כל איבר ברשימה בשורה נפרדת

```
numbers_lst = [10, 20, 30, 40, 50]
for num in numbers_lst:
    print(num)
```

באמצעות ערכים

```
numbers_lst = [10, 20, 30, 40, 50]
for i in range(len(numbers_lst)):
    print(numbers_lst[i])
```

באמצעות אינדקסים



len מחזיר את האורך של הרשימה

מבנה נתונים המאפשר לנו לשמור מספר ערכים, אך הוא שונה מרשימה בכך שאין חשיבות לסדר האיברים לא נוכל לשלוף או לעדכן ערכים באמצעות אינדקסים. בדומה לרשימות הערכים יכולים להיות ממגוון של סוגי משתנים: מחרוזת, מספרים, ערכים בוליאנים ועוד.

את הערכים במילון נגדיר בתוך סוגריים מסולסלים, כל פריט במילון מורכב מצמד הכולל מפתח וערך:

```
my_dict = {"name": "Dan", "age": 25}
```

ערך   מפתח   ערך   מפתח

# מילון – Dictionary – פניה למפתחות ולערכים

```
Print(my_dict.keys())
```

הדפסת כל המפתחות במילון:

```
Print(my_dict.values())
```

הדפסת כל הערכים במילון:

```
my_dict = {"name": "Dan", "age": 25}  
print(my_dict["age"])
```

הדפסת ערך השייך למפתח age:

```
my_dict = {"name": "Dan", "age": 25}  
my_dict["height"] = 1.60
```

הוספת צמד נוסף:

המפתח  
החדש

הערך של  
המפתח  
החדש



מה יודפס על המסך?

```
my_dict = {"name": "Dan", "age": 25}
```

```
my_dict = {"name": "Dan", "age": 25}  
print(my_dict["age"])
```

ד.  
25

ג.  
ערך ריק

ב.  
Age:25

א.  
age

מה יודפס על המסך?

```
my_dict = {"name": "Dan", "age": 25}
```

```
my_dict = {"name": "Dan", "age": 25}  
print(my_dict["age"])
```

א.  
age

ב.  
Age:25

ג.  
ערך ריק

ד.  
25

ניתן לעבור בלולאת for על ערכי המילון

הדפסת כל המפתחות במילון וכל הערכים

מפתח

```
for key in my_dict:  
    print(key, my_dict[key])
```

הערך של  
המפתח

1

כתבו פונקציה המקבלת רשימה של מספרים שלמים ומחזירה את הממוצע שלהם

2

כתבו פונקציה המקבלת רשימה של שמות ושם לחיפוש ובודקת אם השם לחיפוש נמצא ברשימה. במידה וכן היא תחזיר אמת ואת השם. במידה ולא היא תחזיר שקר ואת ההודעה "השם לא נמצא ברשימה."

3

כתבו פונקציה המקבלת מילון. המילון מורכב ממפתחות המתארים מנות במסעדה, וערכי המפתחות יתארו את המחיר של כל מנה. הפונקציה תדפיס את שם המנה היקרה ביותר במסעדה.

לדוגמא – אם המילון הוא:

```
{"burger": 20, "cake": 15, "steak": 50}
```

אז הפונקציה תדפיס את המחרוזת "steak"

# ספריות

**math** ➤

**random** ➤

**ויש עוד...** ➤

ספריות בפייתון הן קבצים המכילים קטעי קוד המוכנים לשימוש.

למשל, בפייתון יש ספריה הנקראת `math` ובה פונקציות המממשות פעולות מתמטיות. לדוגמא, אם נרצה לבצע שורש על מספר, במקום לכתוב קטע קוד משלנו שעושה זאת, נוכל פשוט להשתמש בפונקציה הזו מהספרייה הנ"ל.

```
import math
# sqrt computes the square root
square_root = math.sqrt(4)
print("Square Root of 4 is",square_root)
# pow() computes the power
power = pow(2, 3)
print("2 to the power 3 is",power)
```

### Output

```
Square Root of 4 is 2.0
2 to the power 3 is 8
```

In the above example, we have used

- `math.sqrt(4)` - to compute the square root of 4
- `pow(2, 3)` - computes the power of a number i.e.  $2^3$

```
import random

# הדפסת 10 מספרים בטווח שבין 10 עד 20

for i in range(10):

    print(random.randint(10,20))
```

```
import random

# בחירת שם מרשימה

names = ["Julina", "Mary", "Majd", "Gazal"]

raname = random.choice(names)

print(raname)
```