

פייתון – סיכום שיעור 12

פונקציות

מהי פונקציה?

פונקציה היא קטע קוד שקוראים לו בשם. לקטע הקוד הזה ניתן להעביר מידע שעליו יבוצעו פעולות. קטע הקוד הזה יכול גם להחזיר לנו ערכים חזרה.

למה ליצור פונקציות?

פונקציות מבצעות את אותה סדרת פעולות קבועה ולכן מקילה על כתיבת תוכניות.

איך ניצור פונקציה חדשה?

הצהרה על פונקציה תתבצע באופן הבא:

```
def func_name(param1, ..., paramn):
```

כאשר `def` הוא קיצור של `define`, כלומר אנו מגדירים פונקציה חדשה בשם `func_name` שמקבלת `param1 ... paramn` כפרמטרים לפונקציה.

איך נקרא לפונקציה שייצרנו בפייתון?

כמו שקראנו לפונקציות עד עכשיו:

```
func_name(param1, ..., paramn)
```

כאשר `func_name` היא שם הפונקציה ו `param1 ... paramn` הם פרמטרים לפונקציה.

איך מחזירים ערך?

בעזרת המילה השמורה `return` נוכל להחזיר ערך או ערכים חזרה למי שקרא לפונקציה.

דוגמאות:

```
1 def sum(num1, num2):
2     return num1 + num2
3
4 a = 1
5 b = 2
6 print(sum(a,b))
7
```

שורה 1: הצהרה על פונקציה בשם `sum` שמקבלת שני פרמטרים `num1` ו `num2`.

שורה 2: גוף הפונקציה – החזרה של תוצאת החיבור של שני הפרמטרים.

שורה 4: השמה של הערך 1 לתוך המשתנה `a`.

שורה 5: השמה של הערך 2 לתוך המשתנה `b`.

שורה 6: קריאה לפונקציה `sum` כאשר הפרמטרים הם `a, b` ולכן מה שנקבל חזרה היא תוצאת החיבור ביניהם, ולכן נדפיס 3.

```
1 def my_print():
2     print("Hello World")
3
4 my_print()
5
```



שורה 1: הצהרה על פונקציה בשם *my_print* שלא מקבלת פרמטרים.

שורה 2: גוף הפונקציה – הדפסה של המחרוזת *Hello World*

שורה 4: קריאה לפונקציה *my_print* ניתן לשים לב מצד ימין כי הודפס *Hello World*