



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# תכנות מתקדם בשפת פייתון

גלעד מרקמן





קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# פקודות בסיסיות

משתנים

תנאים

לולאת while

לולאת for



קריית חינוך "פארק המדע"

בית לערכים, למצינות וחדשנות

# משתנים ועוד

- בפייתון אין צורך להגדיר משתנים מראש או לקבוע את סוג המשתנה.
- סוג המשתנה נקבע בהתאם לערך שמוצב בו, וניתן לשנות את סוג המשתנה במהלך חיי התוכנית על ידי הצבת ערך מסוג אחר.
- פקודה מסתיימת בסוף שורה. אסור לסיים בנקודה פסיק.

```
a = 5
print (type(a))
a = 'Word'
print (type(a))
a = True
print (type(a))
a = 2.5
print (type(a))
```

```
[Running] python
<class 'int'>
<class 'str'>
<class 'bool'>
<class 'float'>
```



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

## תנאי if

```
if a > 3:  
    print('a > 3')  
    b = 1  
elif 0 <= a <= 3:  
    print ('a between 0 to 3')  
else:  
    print ('a is Negative')
```

- בהבדל מ C# או Java:
  - אין צורך לשים סוגריים.
  - התנאי מסתיים בנקודותיים.
  - אין סוגריים מסולסלים. בלוק נקבע באמצעות הזחה .tab.
- פקודה מסתיימת בסוף שורה. אין נקודה פסיק.



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# while

```
i = 1
while i < 6:
    print(i)
    i += 1
    if i == 3:
        break # continue
else:
    print("i is no longer less than 6")
```

• התחביר של לולאת while דומה מאוד ל C# או Java, ההבדלים:

- אין סוגריים.
- יש להוסיף נקודותיים.
- אין סוגריים מסוסלים אלא יש להשתמש בהזחה.

• ניתן בסוף הלולאה להוסיף את הפקודה else: אשר תבוצע פעם אחת כאשר התנאי לא יתקיים.

• הפקודה break מאפשר יציאה יזומה מהלולאה

• הפקודה continue מאפשר ל"דלג" איטרציה.



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# לולאת for

```
colors = ['blue', 'Red', 'Green', 'yellow']  
for color in colors:  
    print(color)
```

```
for i in range(1, 8, 2):  
    print(i)
```

```
for i in range(6):  
    print(i)
```

```
1  
3  
5  
7
```

```
0  
1  
2  
3  
4  
5
```

- לולאת for היא למעשה לולאת foreach ונועדה למעבר על מבני נתונים, כגון: List, tuple, dictionary, string.
- המשתנה מקבל בכל איטרציה את הערך של הסדרה.
- על מנת לבצע לולאת for רגילה יש ליצור רשימה של מספרים באמצעות הפונקציה range(start, stop, step).
- Range(n) כול את כל המספרים מ 0 עד :n-1



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# פונקציות

הגדרת פונקציה

פרמטרים



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

```
def function_Name (a , b):  
    print('this is function')  
    return a + b  
  
print (function_Name (3, 5))
```

```
def function (*nums):  
    print ('nums: ', nums)  
    print ('num[2]: ', nums[2])
```

```
nums: (3, 7, 2, 5)  
num[2]: 2
```

# פונקציות

- הגדרת פונקציה מתחילה במילה השמורה .def
- הפונקציה מחזירה ערך אם כוללת את המילה השמורה .return
- ניתן להגדיר פונקציה עם מספר משתנה של ארגומנטים באמצעות \*args. המשתנה שנוצר הוא tuple הכולל את הארגומנטים שהועברו (ראה בהמשך מהו tuple).





קריית חינוך "פארק המדע"

בית לערכים, למצינות וחדשנות

# פונקציות – ארגומנטים עם מפתח

- ניתן להעביר פרמטרים לפונקציה שלא לפי סדר ההגדרה באמצעות מפתחות (שם הארגומנט).

```
def function (a , b):  
    print ('a: ', a)  
    print ('b: ', b)  
  
function (b = 'beta', a = 'alpha')
```

```
a: alpha  
b: beta
```

- נקבל אותה תוצאה בדיוק עם הקריאות הבאות:

```
function ( a = 'alpha', b = 'beta')  
function ('alpha', 'beta')
```



# פונקציה – ברירת מחדל לארגומנט

קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

- ניתן להגדיר לארגומנט ברירת מחדל, כך שאם לא מעבירים פרמטר מתאים ערך ברירה המחדל יתקבל.

```
def Myfunction (a = 'default'):  
    print ('a: ', a)
```

```
Myfunction ()
```

```
a: default
```

- ניתן לקבוע כי חלק מהארגומנטים הם חובה וחלק אופציונאליים:

```
def Myfunction (a, b = 0, c = None):  
    print ('a: ', a, ', b: ', b, ', c: ', c)
```

```
Myfunction ('a')
```

```
a: a , b: 0 , c: None
```

- None דומה לערך null בשפות אחרות.



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# פונקציה – ברירת מחדל לארגומנט

- הערכים ללא ברירת מחדל חייבים להיות ראשוניים.
- באמצעות מפתחות ניתן להעביר פרמטרים שונים לפונקציות.

```
def Myfunction (a, b = 0, c = None):  
    print ('a: ', a, ', b: ', b, ', c: ', c)
```

```
Myfunction ('a')
```

```
Myfunction ('a', 5)
```

```
Myfunction ('a', 5, 9.3)
```

```
Myfunction ('a', c = 9.3)
```

```
a: a , b: 0 , c: None
```

```
a: a , b: 5 , c: None
```

```
a: a , b: 5 , c: 9.3
```

```
a: a , b: 0 , c: 9.3
```