

משתנים בפייתון Python Variables

משתנים הם בתים בזיכרון לאחסון ערכים של נתונים. אפשר גם להגדיר אותם כמיכל לשמירה של ערכים של נתונים.

1. יצירה של משתנים - Creating Variables (הגדרת משתנים)

לפייתון אין פקודה להכריז/להצהיר על משתנה. המשתנה נוצר ברגע שמקצים לו ערך בפעם הראשונה.

לדוגמה :

```
x = 1423
y = "David"
print(x)
print(y)
```

ההדפסה שנקבל :

```
1423
David
```

בפייתון ניתן להדפיס גם בעברית . לדוגמה :

```
z = "דוד המלך"
print(z)
```

ההדפסה שנקבל :

```
דוד המלך
```

אין צורך להצהיר על משתנים עם טיפוס מסוים, וגם ניתן לשנות את הטיפוס שלהם לאחר הגדרתם. לדוגמה :

```
x = 100 # מטיפוס שלם x
x = "Noam" # str ל טיפוס ל
print(x)
```

ההדפסה שנקבל :

```
Noam
```

2. ליהוק - Casting

אם רוצים לציין את טיפוס הנתונים של משתנה, ניתן לעשות זאת עם ליהוק casting. לדוגמה :

```
x = str(3) # x יהיה '3'
y = int(3) # y יהיה 3
z = float(3) # z יהיה 3.0
```

אם לא היינו עושים casting אז המשתנה היה מטיפוס שלם. ה casting שינה את הטיפוס של המשתנה.

```
x = int(7)
y = str(7)
z = float(7)
```

אם נבקש הדפסה ואת טיפוס המשתנה :

```
print(x)
print(y)
print(z)

print (type(x))
print (type(y))
print (type(z))
```

נקבל :

```
7
7
7.0
<class 'int'>
<class 'str'>
<class 'float'>
```

3. קבלת טיפוס הנתון

ניתן לקבל את טיפוס הנתון של משתנה בעזרת הפונקציה `type ()` . לדוגמה:

```
x = 23
y = "David"
print(type(x))
print(type(y))
```

נקבל :

```
<class 'int'>
<class 'str'>
```

4. האם יש לכתוב מחרוזת בין גרשים או בין גרשיים ?

ניתן להצהיר על משתנה בגרשים (' ') או בגרשיים (" ") . לדוגמה:

```
x='David'
y="David"
```

בשני המשתנים יש אותה מחרוזת.

5. Case Sensitive

בשפת פייתון , שלא כמו באסמבלי , יש חשיבות האם התו נרשם באותיות קטנות או גדולות. לדוגמה:

```
A=5
a=10
```

נוצרו כאן 2 משתנים שונים שבכל אחד ערך אחר.

6. שמות של משתנים

שם של משתנה יכול להיות קצר (כמו x או y) או ארוך יותר המתאר את תפקיד המשתנה כמו age לגיל או first_name לתיאור שם פרטי וכו'.

כללים לשמות של משתנים :

- שם משתנה חייב להתחיל עם תו או קו תחתון (_ underscore) .
- שם משתנה איננו יכול להתחיל עם מספר .
- שם משתנה יכול להכיל תווים אלפא נומריים בלבד וקו תחתון (_ , 0-9, A-z) . אין להשתמש בתווים כמו @ \$ % , . : ; / וכו' .
- שמות משתנים הם case-sensitive . name, Name, NAME הם שלושה משתנים שונים .

המשתנים הבאים אינם חוקיים :

```
lname = "David"           # אסור להתחיל עם מספר
my-name = "Moses"        # אסור את התו מינוס ( או מקף )
my name = "EliYahoo"     # אסור רווח בשם המשתנה
```

7. שם משתנה המורכב ממספר מילים

שם של משתנה עם מספר מילים יכול להיות קשה לקריאה. יש מספר טכניקות לקריאות טובה יותר של משתנה המורכב ממספר מילים.

7.א שיטת הגמל

כל מילה חוץ מהמילה הראשונה מתחילה עם אות גדולה.

דוגמאות : firstName, lastName, localTime, timeInDestination .

7.ב שיטת פסקל

כל מילה כולל הראשונה מתחילה באות גדולה.

דוגמאות : FirstName, LastName, LocalTime, TimeInDestination .

7.ג שיטת הנחש

בין כל מילה למילה יש קו תחתון.

דוגמאות : first_name, last_name, local_time, time_in_destination .

8. השמה של ערכים רבים

8.א ניתן לבצע השמה של ערכים למספר משתנים בשורה אחת. לדוגמה :

```
x, y, z = "Red", "Green", "Blue"
```

8.ב ניתן לבצע השמה של ערך אחד למספר משתנים בשורה אחת .

```
x = y = z = "Red"
```

```
num1=num2=num3=100
```

9. פירוק של אוסף (פירוק קבוצה) – unpack a collection

אם יש אוסף של ערכים כמו ב list או tuple ניתן לחלק כל ערך למשתנה. לדוגמה:

```
colors = ["red", "green", "blue"]
x, y, z = colors
print(x)
print(y)
print(z)
```

נקבל:

```
red
green
blue
```

10. הדפסה של משתנים.

כדי לראות את ערכם של משתנים נהוג להשתמש בפקודה `print ()`.

כדי לשלב גם טקסט וגם ערכי משתנים משתמשים בתו `+`. לדוגמה:

```
x="Hello"
print (x+"kita Yud Dalet")
```

ההדפסה שנקבל : Hello kita Yud Dalet

ניתן להשתמש ב `print` כדי לחבר 2 ערכים . לדוגמה :

```
x="Hello"
y="students"
print(x+ " " +y)
```

ההדפסה שנקבל : Hello students

עבור מספרים התו `+` משמש לחיבור. לדוגמה :

```
x, y = 5 , 17.5
print(x+y)
```

ההדפסה שנקבל היא : 22.5

אם מנסים לשלב בין ערך מספרי ומחרוזת נקבל שגיאה ! לדוגמה :

```
x="Hello"
y=5
print (x+y)
```

נקבל : can only concatenate str (not "int") to str

אם נרשום את התוכנית הבאה :

```
y="Hello"
x=5
print (x+y)
```

נקבל הודעה שגיאה : unsupported operand type(s) for +: 'int' and 'str'

11. משתנים גלובליים

משתנים הנוצרים מחוץ לפונקציה נקראים משתנים גלובליים. ניתן להשתמש בהם בתוך פונקציות ומחוץ לפונקציות. לדוגמה:

```
x='Hello'          # func1 מעל הפונקציה
def func1 ( ):    # יצירה/הגדרה של פונקציה
print ( x + ' ' + "Students ")
```

ההדפסה שנקבל: Hello Students

12. משתנה מקומי / לוקלי

משתנה המוגדר בתוך פונקציה הוא משתנה מקומי – local ונקרא גם אוטומטי או פנימי. במשתנה מקומי ניתן להשתמש בפונקציה שבה הוא נוצר/הוגדר בלבד. לדוגמה:

```
Def func ( )
x= 'Hello'      # משתנה מקומי
ניתן ליצור משתנה גלובלי ומשתנה מקומי עם אותו השם. בפונקציה המשתנה המקומי הוא זה
שמטופל. מחוץ לפונקציה מטופל המשתנה הגלובלי. לדוגמה:
```

```
x= "Hello"      # משתנה גלובלי כי הוא לא מוגדר בתוך פונקציה
def func ( ) :  # func בשם
    x=5;        # משתנה מקומי
    print(x)    # בפונקציה יודפס המשתנה המקומי 5
func()         # קריאה לפונקציה
print(x)      # מחוץ לפונקציה יודפס המשתנה הגלובלי Hello
```

ההדפסות שנקבל:

5
Hello

13. שימוש במילת המפתח global

בדרך כלל כשיוצרים משתנה בתוך פונקציה המשתנה הוא מקומי וניתן להשתמש בו בפונקציה בלבד. ניתן ליצור בתוך פונקציה משתנה גלובלי עם מילת המפתח global. במקרה כזה הוא ישמש כמשתנה גלובלי עם טווח הכרה בכל התוכנית מנקודת היצירה כלפי מטה. לדוגמה:

```
def func():
    global x
    x = "Students"
```

הקריאה לפונקציה:

func()

print("Hello " + x)

ההדפסה שנקבל: Hello Students

ניתן להשתמש במילה `global` כאשר רוצים לשנות את ערכו של משתנה גלובלי בתוך פונקציה. לדוגמה:

```
x = "HI"
def func():
    global x
    x = "Hello"

func()
print(x + " Students")
```

נקבל: Hello Students

תרגילים במשתנים

- יש ליצור משתנה בשם `food` ולהעביר לו את שם המאכל האהוב עליך.
 - יש ליצור משתנה בשם `x` ולהעביר לו את הערך 125.
 - מה תהיה ההדפסה עבור התוכנית הבאה?
- ```
x, y = 5, 17.5
print(x+y)
```
- יש לרשום תוכנית שתמצא את ההפרש `x-y`, את המכפלה `x*y` את יחס החלוקה `x/y` ואת  $x^2$ .
  - מי מבין 7 שמות המשתנים הבאים איננו חוקי?
1. Hello 2. Bcd2hex 3. 2hell 4. x\$ 5. xInDollars 6. x\_in-dollars 7. 8. isIt?
- יש לרשום שורת פקודה אחת שתעביר למשתנים `x y z` את הערך 324.
  - יש לרשום שורה אחת שבה נעביר ל `x` את הערך `dollars` ול `y` את הערך `euro`.
  - יש לרשום תוכנית שתעביר ל `x` את הערך 5. לאחר מכן יש ליצור פונקציה בשם `change` שתשנה את ערכו של `x` לגלובלי ל "yes" ותדפיס את הערך החדש.
  - נתונה התוכנית הבאה. מה תהיה ההדפסה כאשר התוכנית רצה?

```
x="Hi"
def func():
 global x
 x = "Students"
func()
print("Hello " + x)
```

10. מה תהיה ההדפסה עבור התוכנית הבאה?

```
x="Hi"
def func():
 global x
 x = "Students"
print("Hello " + x)
func()
```

## פתרון התרגילים במשתנים

food = " Huge Stake" .1

x=125 .2

22.5 .3

.4

```
print (x-y)
print (x*y)
print(y/x)
print(x*x)
```

7 6 4 3 המשתנים במקומות .5

x=y=z=324 .6

x, y = "dollars", "euro" .7

.8

```
x = 5
def change () :
 global x
 x="yes"
```

```
change() # קריאה לפונקציה כדי ש x ישנה את ערכו
print(x)
```

Hello Students .9

Hello Hi .10