

מהו יצרן או באנגלית generator בשפת פייתון?

זוהי פונקציה שמחזירה עצם שעליו ניתן להפעיל את הפונקציה `next()`, בצורה כזו שבכל קריאה היא תחזיר את האיבר הבא (ערך כלשהו שנוצר בתוכה בעזרת פקודת `yield`) עד סיום הערכים ואז תתקבל הודעת שגיאה מסוג: `StopIteration.exception`.
האובייקט שנוצר הוא מסוג שניקרא איטרטור.

פונקציה רגילה מחזירה ערך / ערכים בסיום פעולתה על ידי משפט `return`. פונקציה מסוג generator משתמשת במילת המפתח `yield`, לייצר ערך (יכול להופיע בכל מקום בפונקציה)

דוגמה:

כפי שרואים `myGen(n)` היא פונקציה שמייצרת את n ואת $n+1$. כל קריאה לפונקציה `next` מייצרת ערך בודד עד שנגמרים הערכים. כדי לעבוד עם לולאה, `next` נקראת ברקע, למשל:

```
>>> def myGen(n):  
...     yield n  
...     yield n + 1  
...  
>>> g = myGen(6)  
>>> next(g)  
6  
>>> next(g)  
7  
>>> next(g)  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
StopIteration
```

כפי שרואים `myGen(n)` היא פונקציה שמייצרת את n ואת $n+1$. כל קריאה ל `next` מייצרת ערך בודד עד שנגמרים הערכים. כדי לעבוד עם לולאה, `next` נקרא ברקע, למשל:

```
>>> for n in myGen(6):  
...     print(n)  
...  
6  
7
```

המטרה המרכזית של פונקציה מסוג generator זה לחסוך מקום בזיכרון, למשל אם ישנה סידרה גדולה של מספרים וניתן לייצר כל ערך ללא תלות בכל הערכים שהיו קודם (למשל פיבונאצ'י), לא צריך לשמר את הסידרה בזיכרון, רק לייצר ולקרוא לה בעת הצורך.

```
def genfibon(n):  
    """  
    Generate a fibonnaci sequence up to n  
    """  
    a = 1  
    b = 1  
    for i in range(n):  
        yield a
```

```
for num in getfibon(10):  
    print (num)
```

```
1  
1  
2  
3  
5  
8  
13  
21  
34  
55
```

עם פונקציה רגילה זה היה נראה כך:

```
def fibon(n):  
    a = 1  
    b = 1  
    output = []  
  
    for i in range(n):  
        output.append(a)  
        a,b = b,a+b  
  
    return output
```

כעת מראה מה הפונקציה `iter()` עושה.

לשם כך נראה דוגמה. אנו יודעים שאפשר לעבור על מחרוזת כסדרת תוים למשל:

```
st1 = 'hello'
for letter in st1:
    print(letter)
```

```
h
e
l
l
o
```

זה אינו אומר שמחרוזת היא איטרטור. למשל:

```
next(st1)
```

TypeError: 'str' object is not an iterator

אז למרות שמחרוזת היא סידרת תוים איננו יכולים להתייחס אליה כאיטרטור אלא אם נעשה את הדבר הבא:

```
s_iter = iter(st1)
next(s_iter)
'h'
next(s_iter)
'e'
```

כלומר הפכנו את המחרוזת לאיטרטור בעזרת הפונקציה `iter()`