

## Tutoriat 6

### Lambda Funcții:

- Sunt funcții anonime.
- Pot lua orice număr de argumente.
- Pot avea o singură expresie.
- Sunt adesea folosite la sortarea listelor.

### Definire:

lambda <argumente> : <expresie>

### Ex:

Cod	Rezultat
<pre>x = lambda x,y,z : x+y+z+10 print(x(5,1,2))</pre>	18
<pre>x = lambda x,y=0,z=0 : x+y+z+10 print(x(5))</pre>	15
<pre>x = lambda x,y=0,z=0 : x+y+z+10 print(x(5,z=3))</pre>	18
<pre>#f este o funcție ce returnează o lambda funcție def f(a,b,c):     return lambda x : a*(x**2) + b*x + c  polinom = f(2,5,11) #polinom este o funcție ce                     #returne calculeaza 2x^2 +5x+11  print(polinom(2))</pre>	29
<pre>x = lambda x,y=0,z=0 : x+y+z+10 print(x)</pre>	<function <lambda> at 0x000001FF32E63E20>

## Recursivitate

### Formă:

```
def <numeFunctie> (<parametrii>):  
    if <condiție de oprire>==False :  
        <bloc instrucțiuni>  
        <numeFunctie> (<parametrii>)  
        <bloc instrucțiuni>  
    else:  
        <bloc instrucțiuni>
```

### Observații:

- În python este definită o limita superioara de autoapelari ale functiei. Dacă este depasita se generează o eroare.

Ex:

def F(): F()  F()	File "c:\Users\test.py", line 2, in F F() [Previous line repeated 996 more times]
----------------------------	---

- Aceasta limita este implementată pentru a evita eroarea **Stack Overflow**, care poate apărea și dacă nu este o recursivitate infinită.
- Python folosește o adâncime maximă de recursivitate de 1000.
- Pentru a modifica adancimea maximă de recursivitate (deși nu este indicat) putem folosi funcția **setrecursionlimit()**

### Ex:

Cod	Rezultat
<pre>def factorial(x):     if x == 1: #conditie de oprire         return 1     else:         return (x * factorial(x-1))  print(factorial(5))</pre>	120

## Sortari de liste

### Funcția sort:

- Este similară cu funcția sorted().
- Complexitate:  $O(n \log n)$ .

### Parametrii:

- reverse, by default este False, dacă este True va sorta lista “descrescător” ;
- key, primește o funcție care servește drept criteriu de comparație;

### Exemple:

Cod	Rezultat
<pre>l=['ana','mere','gutui','avea','si',':'] l.sort(key=len) print(l)</pre>	<pre>[':', 'si', 'ana', 'mere', 'avea', 'gutui']</pre> <p>Obs: sortează după lungime</p>
<pre>l=['ab','ac','a','za','wwa','ca'] l.sort() print(l)</pre>	<pre>['a', 'ab', 'ac', 'ca', 'wwa', 'za']</pre> <p>Obs: sortează lexicografic</p>
<pre>l=['ab','ac','a','za','wwa','ca'] l.sort(key=lambda x:x[0]) print(l)</pre>	<pre>['ab', 'ac', 'a', 'ca', 'wwa', 'za']</pre> <p>Obs: sortează după primul caracter</p>
<pre>def f(x):     nr=0     for chr in x:         if chr in "aeiouAEIOU":             nr+=1     return nr  l=['ab','adewc','a','zaeas','wwaa','ca'] l.sort(key=f) print(l)</pre>	<pre>['ab', 'a', 'ca', 'adewc', 'wwaa', 'zaeas']</pre> <p>Obs: sortează după nr de vocale.</p>