

Programare I

Curs 2. Partea 2. Structuri de date în Python. Instrucțiuni repetitive.

Botescu Mihai
mihai.botescu00@e-uvt.ro

April 2021

```
In [1]: l = []
In [2]: l
In [3]: type(l)
In [4]: t = tuple()
In [5]: t
In [6]: type(t)
In [7]: [1,2,3]
In [8]: (1,2,3)
In [9]: x, y, z = 2,3,4
In [10]: x
In [11]: y
In [12]: z
In [13]: (x,y,z) = (2,3,4)
In [14]: x
In [15]: y
In [16]:
In [16]: z
In [17]: x,y,z = 2,3
#eroare. de ce? explicați!
In [18]: len(t)
In [19]: t = tuple()
In [20]: t
```

```

In [21]: t = (1,2,3)

In [22]: len(t)

In [23]: t[0]

In [24]: t[1]

In [25]: (1,2,3) + (3,4)

In [26]: (1,2,3) * 0

In [27]: (1,2,3) * 2

In [28]: (1,2,3) * (1,2)

In [29]: i,j = 0, 2

In [30]: i

In [31]: j

In [32]: t[i::j]

In [33]: t[i:j]

In [34]: t[i:j:]

In [35]: t[i:j:1]

In [36]: t[i:j:2]

In [37]: t

In [38]: t = (1,2,3,4,5,6,7)

In [39]: t

In [40]: t[0::2]

In [41]: # t[0::2] ---> t[0], t[2], t[4], t[6]

In [42]: t[0::-1]

In [43]: t[::-1]

In [44]: for x in t:
...:     print("x inmultit cu 2:", 2*x)
...:
x inmultit cu 2: 2
x inmultit cu 2: 4
x inmultit cu 2: 6
x inmultit cu 2: 8
x inmultit cu 2: 10
x inmultit cu 2: 12

```

```

x inmultit cu 2: 14

In [45]: l1 = [5]

In [46]: l2 = l1

In [47]: l1

In [48]: l2

In [49]: l1.append(6)

In [50]: l1

In [51]: l2

In [52]: l2 = l1[:]

In [53]: l2

In [54]: l1.append(2)

In [55]: l1

In [56]: l2

In [57]: matrice = [[1,0,0],[0,1,0],[0,0,1]]

In [58]: matrice

In [59]: for linie in matrice:
...:     for coloana in matrice:
...:         print(coloana,end= ' ')
...:     print()
...:
[1, 0, 0] [0, 1, 0] [0, 0, 1]
[1, 0, 0] [0, 1, 0] [0, 0, 1]
[1, 0, 0] [0, 1, 0] [0, 0, 1]

In [60]: for linie in matrice:
...:     for coloana in linie:
...:         print(coloana,end= ' ')
...:     print()
...:
1 0 0
0 1 0
0 0 1

In [61]: 

In [61]: matrice

In [62]: for i in range(len(matrice)):
...:     for coloana in range(len(linie)):
...:         print(matrice[i][j],end= ' ')
...:     print()
...:

```

```

0 0 0
0 0 0
1 1 1

In [63]: for i in range(len(matrice)):
...:     for coloana in range(len(matrice[i])):
...:         print(matrice[i][j],end=' ')
...:     print()
...:
...:
0 0 0
0 0 0
1 1 1

In [64]: for i in range(len(matrice)):
...:     for j in range(len(matrice[i])):
...:         print(matrice[i][j],end=' ')
...:     print()
...:
...:
1 0 0
0 1 0
0 0 1

In [65]: s = set()

In [66]: s

In [67]: s = {1,2,3,4,5}

In [68]: s

In [69]: s.add(2)

In [70]: s

In [71]: s.add(2.5)

In [72]: s

In [73]: len(s)

In [74]: s.union({-1,2.5,1,4})

In [75]: s

In [76]: s.intersection({-1,2.5,1,4})

In [77]: s

In [78]: s.difference({-1,2.5,1,4})

In [79]: s1 = {1,2,3,4}

In [80]: s1.intersect({5,6,7})
#eroare. de ce?

```

```
In [81]: s1.intersection({5,6,7})  
In [82]: sq1  
-----  
NameError                                     Traceback (most recent call last)  
<ipython-input-82-4bab670be481> in <module>  
----> 1 sq1  
  
NameError: name 'sq1' is not defined  
  
In [83]: s1  
  
In [84]: d = dict()  
  
In [85]: type(d)  
  
In [86]: d  
  
In [87]: d["ana"] = 1  
  
In [88]: d  
  
In [89]: d["ana"] = []  
  
In [90]: # nu avem voie valori separate in dictionar  
  
In [91]: d["ana"].append('maria')  
  
In [92]: d  
  
In [93]: d["ana"].append(2)  
  
In [94]: d  
  
In [95]: d[2] = 'anamaria'  
  
In [96]: d  
  
In [97]: d[2] = ['ion','marcel']  
  
In [98]: d  
  
In [99]: d[2].append([1,2,3,4,5])  
  
In [100]: d  
  
In [101]: d.keys()  
  
In [102]: for cheie in d.keys():  
...:     print(cheie)  
...:  
ana  
2  
  
In [103]: for valoare in d.values():  
...:     print(valoare)  
...:
```

```

['maria', 2]
['ion', 'marcel', [1, 2, 3, 4, 5]]

In [104]: 

In [104]: for cheie, valoare in d.items():
    ...:     print(f'Cheie : {cheie}; valoare: {valoare}')
    ...:
Cheie : ana; valoare: ['maria', 2]
Cheie : 2; valoare: ['ion', 'marcel', [1, 2, 3, 4, 5]]

In [105]: def prim(x):
    ...:     # un numar prim este un numar divizibil cu 1 si cu el insusi.
    ...:     # ce cautam ? divizorii in afara de 1 si el insusi.
    ...:     d = 2
    ...:     while d <= x//2:
    ...:         if x % d == 0: #restul impartirii lui x la d este 0
    ...:             print('NU ESTE PRIM')
    ...:             return None
    ...:         d = d + 1
    ...:     print('ESTE PRIM')
    ...:

In [106]: prim(2)
ESTE PRIM

In [107]: prim(4)
NU ESTE PRIM

In [108]: prim(5)
ESTE PRIM

In [109]: prim(10)
NU ESTE PRIM

In [110]: prim(13)
ESTE PRIM

In [111]: prim(173)
ESTE PRIM

In [112]: n = 123

In [113]: n

In [114]: s = 0

In [115]: s

In [116]: while n != 0:
    ...:     s = s + (n % 10)
    ...:     n = n//10 #tai cifrele lui n
    ...:

In [117]: s

In [118]: def cea_mai_frecventa(n):

```

```

....:     cifre = dict() #numar toate aparitiile cifrelor lui n => CHEIA : CIFRA LUI N, VALOAREA :
....:     while n != 0:
....:         if (n%10) not in cifre.keys():
....:             cifre[n%10] = 1 #marchez prima aparitie a unei cifre
....:         else:
....:             cifre[n%10] = cifre[n%10] + 1 #altfel: cifra apare deja. maresc cu 1 numarul de a
....:             maxim = 0
....:             cifra = int()
....:             for cf, aparitii in cifre.items():
....:                 if aparitii > maxim:
....:                     maxim = aparitii
....:                     cifra = cf
....:             return (cifra, maxim)
....:
....:

```

In [119]: cea_mai_frecventa(12123)

^{^C-----}

KeyboardInterrupt

Traceback (most recent call last)

```
<ipython-input-119-4dc22f31683d> in <module>
----> 1 cea_mai_frecventa(12123)
```

```
<ipython-input-118-86f5fa25c687> in cea_mai_frecventa(n)
    5         cifre[n%10] = 1 #marchez prima aparitie a unei cifre
    6     else:
----> 7         cifre[n%10] = cifre[n%10] + 1 #altfel: cifra apare deja. maresc cu 1 numarul de a
    8     maxim = 0
    9     cifra = int()
```

KeyboardInterrupt:

#de ce while-ul de mai sus ciclează la infinit? explicați!

In [120]: **def** cea_mai_frecventa(n):

```

....:     cifre = dict() #numar toate aparitiile cifrelor lui n => CHEIA : CIFRA LUI N, VALOAREA :
....:     while n != 0:
....:         if (n%10) not in cifre.keys():
....:             cifre[n%10] = 1 #marchez prima aparitie a unei cifre
....:         else:
....:             cifre[n%10] = cifre[n%10] + 1 #altfel: cifra apare deja. maresc cu 1 numarul de a
....:             n = n // 10
....:             maxim = 0
....:             cifra = int()
....:             for cf, aparitii in cifre.items():
....:                 if aparitii > maxim:
....:                     maxim = aparitii
....:                     cifra = cf
....:             return (cifra, maxim)
....:
....:
```

In [121]: cea_mai_frecventa(12123)

In [122]: **def** cea_mai_frecventa(n):

```

....:     cifre = dict() #numar toate aparitiile cifrelor lui n => CHEIA : CIFRA LUI N, VALOAREA :
....:     while n != 0:
....:         if (n%10) not in cifre.keys():
....:             cifre[n%10] = 1 #marchez prima aparitie a unei cifre
```

```
....:     else:
....:         cifre[n%10] = cifre[n%10] + 1 #altfel: cifra apare deja. maresc cu 1 numarul de a
....:         n = n // 10
....:     maxim = 0
....:     cifra = int()
....:     for cf, aparitii in cifre.items():
....:         if aparitii > maxim:
....:             maxim = aparitii
....:             cifra = cf
....:     return (cifre, cifra, maxim)
....:
....:
....:
```

In [123]: cea_mai_frecventa(12123)

In [124]: "anamaria"

In [125]: # d = { 'a': 4, 'n' : 1, 'm': 1, 'r': 1, 'i': 1}

In [126]: ()

In [127]: type(())

In [128]: type({})