

Tutoriat 3

Liste

Liste:

-sunt un tip de date iterabil, ce pot stoca tipuri diferite de date.

Ex: lista= [1,'2',[3,'ána']]

-pot fi modificate după declarare.

Metode de declarare:

Exemple:

folosind []:

l1=[] (lista goala)

l2 = [1,2,3,4,5]

folosind constructorul list()

list(<iterabil>)

l3=list("1234") => l3 = ['1', '2', '3', '4']

folosind *

l4= <lista> * n => l4 este format prin concatenarea lui <lista> de n ori , dacă n este negativ sau 0 atunci se va memora []

Ex:

l = [1,2,3] * 3 => l=[1,2,3,1,2,3,1,2,3]

Parcurgerea listelor:

Metoda 1

for <variabila> in <lista>:

print(<variabila>)

Metoda 2

for <variabila> in range(len(<lista>))

print(<lista>[<variabila>])

Slicing:

Tip accesare	Rezultat
s[i]	Elementul de pe poziția i în lista
s[i:]	returnează o listă ce conține elementele începând cu poziția i în s pana la finalul șirului
s[i:j]	returnează o listă ce conține elementele

	incepand cu poziția i în s pana la poziția j-1
s[:]	returnează întreaga secvență
s[:j]	returnează o listă ce conține elementele începand cu poziția 0 în s până la poziția j-1
s[::-1]	returnează întreaga secvență în ordine inversă Ex: [1,2,3,4][::-1]=>[4,3,2,1]

Metodele:

Metoda	Descriere
<lista>.append(<valoare>)	Adaugă <valoare> la sfârșitul listei
<lista>.copy()	Returnează o copie a listei
<lista>.count(<val>)	Returnează numărul de elemente cu valoarea specificată
<lista>.extend(<iterabil>)	Adăugați elementele unei iterabil, la sfârșitul listei curente
<lista>.sort()	Sortează lista
len(<lista>)	Returnează lungimea listei
<lista>.insert(<poziție>,<valoare>)	Adaugă un element la poziția specificată
<lista>.pop(<poziție>)	Îndepărtează elementul din poziția specificată !dacă poziția e mai mare decât lungimea listei => eroare.
<lista>.remove(<valoare>)	Îndepărtează elementul cu valoarea specificată !dacă valoarea nu exista in lista => eroare

! Pentru a verifica dacă o lista 'l' conține valoarea 'val' putem folosi **in**

Ex:

```
if val in l:
    print("Exista")
else:
    print("Nu exista")
```

Concatenarea de liste:

Se poate folosi operatorul + , dar și funcțiile append si extend

Ex:

```
l1 = [1,2,3,4] + [2,3] => [1,2,3,4,2,3]
```

```
l2=[1,2,3,4]
```

```
l2.extend("1234") => l2=[1, 2, 3, 4, '1', '2', '3', '4']
```

```
l3=[1,2,3,4]
```

```
l3.append("1234") => l3=[1, 2, 3, 4, '1234']
```

Copierea listelor:

!!!! Se folosește funcția copy()

Ex:

```
l=[1,2,3]
```

```
l2=l.copy()
```

```
l3=l
```

```
l[1]=10
```

```
print(l) => [1,10,3]
```

```
print(l2) => [1,2,3]
```

```
print(l3) => [1,10,3]
```

Modificarea listelor:

```
l=[1,2,3,4,5]
```

```
l[1:3]=[60] => l = [1, 60, 4, 5]
```

```
l[1:2]=[2,3,7,8] => l = [1, 2, 3, 7, 8, 4, 5]
```

```
l[1]=0 => l = [1, 0, 3, 7, 8, 4, 5]
```

```
l.insert(0,20) => l = [20, 1, 0, 3, 7, 8, 4, 5]
```

```
l.sort() => l = [0, 1, 3, 4, 5, 7, 8, 20]
```

Comprehensiune:

```
l=[<valoare/variabila> for <variabila> in <iterabil> if <condiție>]
```

Exemple:

```
l1=[0 for x in range(4)] => l1=[0,0,0,0]
```

```
l2=[x for x in [1,2,3,4,5] if x%2 == 0] => l2 = [2,4]
```

```
l3=[[x] for x in [1,2,3,4,5] if x%2 == 0] => l3 =[[2], [4]]
```

Funcția de sortare sort():

```
<lista>.sort(reverse=True(False), key=<funcție>)
```

Parametrii reverse și key sunt opționali, default reverse este False.

Cu ajutorul parametrului key putem sa ne definim propriile condiții de comparare.

Complexitatea este O(nlogn)

Exemplu:

```
def functie(element):
    return len(element)

lista = ["leu", "masina", "pisica"]

lista.sort(reverse=True, key=functie)

print(lista) => ['masina', 'pisic', 'leu']
```

Matrice:

Sunt liste de liste.

Exemple creare matrice cu n linii și m coloane:

```
l=[[0]*m for x in range(n)] => l = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
l=[[0 for x in range(m)] for x in range(n)]
=> l = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
```

așa nu

```
l=[[0]*m]*n => l = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
l[0][0]=3 => l = [[3, 0, 0, 0], [3, 0, 0, 0], [3, 0, 0, 0]]
```

```
l=[[0 for x in range(m)]]*n => l = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
l[0][0]=3 => l = [[3, 0, 0, 0], [3, 0, 0, 0], [3, 0, 0, 0]]
```

