

NOTIȚE CURS 4:

LISTE - noutăbilită, indexată de la 0, poate avea val. neomogene, caracter dinamic

1. inițializare

„de mână”

list comprehension

$L = [x \text{ for } x \text{ in range}(10)]$

$L = [x^{**2} \text{ for } x \text{ in range}(10) \text{ if } x \% 2 == 0]$

$L = [x^{**2} \text{ if } x \% 2 == 0 \text{ else } -x^{**2} \text{ for } x \text{ in range}(10)]$

2. accesare

→ prin indici pozitivi (0, n-1) sau negativi (-1, -n)

3. ștergere

→ prin **del L[i]** sau **L.pop()**

4. operatori:

+ = concatenare $[1,2] + [3,4] = [1,2,3,4]$

* = multiplicare $[1,2] * 3 = [1,2,1,2,1,2]$

<, <=, >, >=, ==, != (lexicografic)

5. funcții

len(lista) → nr. de elem. din listă

list(seco) → transformă o listă : $list('test') = ['t', 'e', 's', 't']$

min, max

sum(lista)

sorted(lista, reverse=True/False) → lista nu va fi modificată ci se transformă într-o nouă

etc.

6. metode

L.count(elem) → de câte ori apare elem. în L

L.append(val) → o adaugă la final

L.extend(seco) → adaugă mai multe elem. la final

L.insert(poz, val) → inserează elem. la poz. dată

L.index(val) → prima ap. a valorii sau eroare

L.remove(val) → șterge prima apariție a valorii sau eroare

L.pop() → șterge val. de pe poz. „poz”

L.clear() = $L = []$

L.reverse() oglinditue listei

6. `sort()`

- sortează lista implicit cresc. 1 se poate și descresc. cu `reverse=True`

7. crearea listelor:

cel mai eficient cu x for x in `range(n)`
&
metoda `append()`

8. crearea matricelor

listă de liste

! pt. a duplica o listă nu putem atribui $B=A$ pt. că de fapt se transmite doar referința)
putem totuși cu `B=A.copy()`

↳ shallow copy (dacă A are referințe în listă acestea vor fi copiate și orice modificare va apărea și în B)

`B = copy.deepcopy(A)`

→ rezolvă pb. anterioară
→ are complexitate mai mare

TUPLEURI:

- imutabile, indexată de la 0, poate avea val. neomogene
despachetare + împachetare

1. creare:

`t = ()`

`t = (1,)` ← fără virgulă t va fi 1, int-ul 1

`t = tuple(ceva)`

`t = 2, 3, "mama"`

`t = (2, 3, "mama")`

cu comprehensions

cu `corol` + ↓

2. accesare:

prin index poz. și negativi

3. operatori:

`+` = concatenare

`*` = multiplicare

in, not in

op. relationali (lexicografic)

4. funcții

→ exact ca la liste

5. metode

→ tuple.count și tuple.index