INFORMATICĂ · PYTHON · CLASA A VIII-A · TABLOURI UNIDIMENSIONALE · LECȚIA 2

Ce este un șir de numere? Lista în Python

Durată: 50 min Nivel: gimnaziu · Python Înainte: lecțiile py8g-01

CE VEI ȘTI DUPĂ LECȚIE

- Înțelegi ce este un șir de numere (vector) și de ce avem nevoie de el.
- Creezi o listă în Python și citești un element după indice.
- Știi că indicii încep de la 0 și folosești len().

1. De ce nu ajung variabilele?

Vrei să ții minte notele a 30 de colegi. Ai scrie `nota1`, `nota2`, ... `nota30`? Ar fi un coșmar. Soluția: **un singur nume pentru mai multe valori**. Un astfel de șir de numere se numește **vector** (sau **tablou unidimensional**). În Python îl facem cu o **listă**.

2. Prima listă

```
note = [9, 7, 10, 6, 8]
print(note)
```

```
[9, 7, 10, 6, 8]
```

Lista se scrie între paranteze drepte, cu valorile despărțite prin virgulă. Gândește-te la ea ca la un **șir de căsuțe numerotate**:

indice	0	1	2	3	4
valoare	9	7	10	6	8

LA BAC

Numărătoarea căsuțelor începe de la **0**, nu de la 1. Prima valoare este `note[0]`, iar ultima dintre cele 5 este `note[4]`.

3. Citim o valoare după indice

Numele listei și indicele între paranteze drepte: `note[2]`.

```
note = [9, 7, 10, 6, 8]
print(note[0])
print(note[2])
print(note[2] + note[3])
```

```
9
10
16
```

Ai multe feluri de a crea o listă:

Scrii	Rezultat
`v = [3, 1, 4]`	lista cu 3 valori
`v = []`	listă goală
`v = [0] * 5`	`[0, 0, 0, 0, 0]`

4. Câte elemente are lista? len()

`len(v)` (de la *length*, lungime) spune câte elemente are lista. Atenție: dacă `n = len(v)`, ultimul indice este `n - 1`.

```
v = [5, 8, 2, 7]
n = len(v)
print(n)
print(v[n - 1])
```

```
4
7
```

ATENȚIE — GREȘELI FRECVENTE

`v[4]` la o listă cu 4 elemente nu există. Python se oprește cu eroarea `IndexError: list index out of range`. Înseamnă că ai cerut o căsuță care nu există.

5. Schimbăm o valoare

Elementele unei liste se pot modifica: `v[i] = valoare`.

```
puncte = [10, 20, 30]
puncte[1] = 25
puncte[2] = puncte[2] + 5
print(puncte)
```

```
[10, 25, 35]
```

6. Parcurgem lista

Metoda 1: după indice. `i` ia pe rând valorile 0, 1, ..., `n - 1`.

```
v = [4, 9, 2]
for i in range(len(v)):
    print("v[", i, "] =", v[i])
```

```
v[ 0 ] = 4
v[ 1 ] = 9
v[ 2 ] = 2
```

Metoda 2: după valoare. `x` ia pe rând fiecare element, fără indice.

```python
v = [4, 9, 2]
for x in v:
    print(x * 2, end=" ")
```

```
8 18 4
```

Prima metodă o folosim când avem nevoie de **poziție** sau vrem să **schimbăm** elemente. A doua este bună când ne trebuie doar valorile.

7. Verifică-te

1. Fie `v = [6, 1, 9, 4]`. Cât este `v[1]`? Dar `v[3]`?
2. Câte elemente are `v = [2, 2, 2]`? Care este ultimul indice?
3. Ce afișează `v = [3, 5]` urmat de `v[0] = 8` și `print(v)`?
4. Ce eroare apare la `print(v[2])`, dacă `v = [3, 5]`?
5. Scrie o listă cu 6 zerouri într-un singur rând.

Răspunsuri

1. `1` și `4`.
2. 3 elemente; ultimul indice este 2.
3. `[8, 5]`.
4. `IndexError: list index out of range`: lista are indicii 0 și 1.
5. `v = [0] * 6`.