INFORMATICĂ · PYTHON · CLASA A VIII-A · TABLOURI UNIDIMENSIONALE · LECȚIA 3

Citirea și afișarea unui șir

Durată: 50 min Nivel: gimnaziu · Python Înainte: lecțiile py8g-02

CE VEI ȘTI DUPĂ LECȚIE

- Citești un șir de n numere, fie câte unul pe rând, fie pe un singur rând.
- Afișezi un șir în ordinea dată, în ordine inversă și cu indicii.
- Recunoști formatul „pe primul rând n, pe al doilea cele n numere”.

1. Cum arată datele

La multe probleme primești datele așa:

```
5
8 3 9 1 6
```

Pe primul rând este `n` (câte numere sunt), iar pe al doilea rând sunt cele `n` numere. Programul tău trebuie să le citească și să le țină minte într-o listă.

2. Citire: câte un număr pe rând

Pornim cu lista goală și adăugăm la sfârșit fiecare număr citit cu `append`.

```
n = int(input())
v = []
for i in range(n):
    x = int(input())
    v.append(x)
print(v)
```

Tastezi `3`, apoi `7`, `2`, `5` (fiecare pe rândul lui):

```
[7, 2, 5]
```

`v.append(x)` pune valoarea `x` la sfârșitul listei. Lista crește cu o căsuță.

3. Citire: toate numerele pe același rând

Dacă numerele sunt scrise pe un singur rând, folosim o „formulă” scurtă:

```
n = int(input())
v = list(map(int, input().split()))
print(v)
```

Tastezi `4`, apoi `8 3 9 1`:

```
[8, 3, 9, 1]
```

Cum funcționează, pe rând:

Pas	Ce face	Rezultat pentru `8 3 9 1`
`input()`	citește rândul	`"8 3 9 1"` (text)
`.split()`	taie textul la spații	`["8", "3", "9", "1"]` (bucăți de text)
`map(int, ...)`	transformă fiecare bucată în număr	numerele 8, 3, 9, 1
`list(...)`	face din ele o listă	`[8, 3, 9, 1]`

> **SFAT**
>
> Nu trebuie să înțelegi `map` acum. Copiază formula `list(map(int, input().split()))` ori de câte ori numerele sunt pe același rând.

4. Afișare: un element după altul

Cel mai des se cere afișarea numerelor pe un rând, separate prin spațiu.

```python
v = [8, 3, 9, 1]
for x in v:
    print(x, end=" ")
```

```
8 3 9 1
```

Mai există o scurtătură: `print(*v)` scrie elementele separate prin spațiu. Steluța „desface” lista. Fără ea, `print(v)` afișează și parantezele: `[8, 3, 9, 1]`.

```python
v = [8, 3, 9, 1]
print(*v)
print(*v, sep=", ")
```

```
8 3 9 1
8, 3, 9, 1
```

5. Afișare în ordine inversă

Vrem de la ultimul element la primul. Indicii merg de la `n - 1` până la `0`, deci pasul este `-1`:

```python
v = [8, 3, 9, 1]
n = len(v)
for i in range(n - 1, -1, -1):
    print(v[i], end=" ")
```

```
1 9 3 8
```

`range(n - 1, -1, -1)` se oprește **înainte** de `-1`, deci ultimul indice luat este `0`.

6. Afișare cu poziția

Când ai nevoie și de indice, parcurgi după `i`:

```
v = [8, 3, 9, 1]
for i in range(len(v)):
    print("poziția", i, "are valoarea", v[i])
```

```
poziția 0 are valoarea 8
poziția 1 are valoarea 3
poziția 2 are valoarea 9
poziția 3 are valoarea 1
```

ATENȚIE — GREȘELI FRECVENTE

Dacă ai citit câte un număr pe rând, dar programul așteaptă toate numerele pe același rând (sau invers), apare `ValueError`. Citește cu atenție cum sunt date numerele.

7. Verifică-te

1. Ce înseamnă `v.append(5)`?
2. Cum citești numerele `4 7 1` (pe un rând) într-o listă?
3. Ce afișează `print(*[1, 2, 3])`?
4. Scrie bucla care afișează în ordine inversă un șir cu `n` elemente.
5. De ce nu merge `print(*v)` dacă uiți steluța? (Ce se afișează?)

Răspunsuri

1. Adaugă valoarea 5 la sfârșitul listei `v`.
2. `v = list(map(int, input().split()))`.
3. `1 2 3`.
4. `for i in range(n - 1, -1, -1): print(v[i], end=" ")`.
5. Merge, dar afișează și parantezele și virgulele: `[1, 2, 3]`.