INFORMATICĂ · PYTHON · CLASA A VIII-A · TABLOURI UNIDIMENSIONALE · LECȚIA 8

Interschimbarea, inversarea și rotirea unui șir

Durată: 50 min Nivel: gimnaziu · Python Înainte: lecțiile py8g-07

CE VEI ȘTI DUPĂ LECȚIE

- Interschimbi două variabile sau două elemente ale unui șir.
- Inversezi un șir (primul devine ultimul).
- Rotești un șir cu o poziție.

1. Schimbăm două valori între ele

Ai două pahare: unul cu suc, altul cu apă. Cum le schimbi conținutul? Ai nevoie de **un al treilea pahar**. La fel în program:

```
a = 5
b = 8
aux = a
a = b
b = aux
print(a, b)
```

```
8 5
```

ATENȚIE — GREȘELI FRECVENTE

Dacă scrii direct `a = b` și apoi `b = a`, pierzi valoarea inițială a lui `a`: ambele devin 8. De aceea păstrăm valoarea într-o variabilă ajutătoare, `aux`.

Python permite și o variantă scurtă, pe un singur rând:

```
a, b = 5, 8
a, b = b, a
print(a, b)
```

```
8 5
```

Ambele variante fac același lucru. Mai jos folosim varianta scurtă.

2. Interschimbăm două elemente ale șirului

Funcționează la fel, doar că în loc de variabile avem căsuțe: `v[i], v[j] = v[j], v[i]`.

```
v = [4, 9, 1, 7]
v[0], v[3] = v[3], v[0]
print(*v)
```

```
7 9 1 4
```

Primul și ultimul element și-au schimbat locurile.

3. Inversarea șirului

Inversat, șirul `1 2 3 4 5` devine `5 4 3 2 1`. Schimbăm între ele **perechile de la capete**: primul cu ultimul, al doilea cu penultimul, și așa mai departe. Ne oprim la mijloc.

```
v = [1, 2, 3, 4, 5]
n = len(v)
for i in range(n // 2):
    v[i], v[n - 1 - i] = v[n - 1 - i], v[i]
print(*v)
```

```
5 4 3 2 1
```

Perechea lui `v[i]` este `v[n - 1 - i]`. Pentru `n = 5`, bucla rulează de `5 // 2 = 2` ori:

`i`	Perechea	Lista după schimb
0	`v[0]` cu `v[4]`	`5 2 3 4 1`
1	`v[1]` cu `v[3]`	`5 4 3 2 1`

Elementul din mijloc (`3`) rămâne pe loc, deci nu are nevoie de pereche.

> **ATENȚIE — GREȘELI FRECVENTE**
>
> Dacă ai parcurge toată lista (`range(n)`), ai schimba fiecare pereche **de două ori**, deci șirul ar reveni la forma inițială. De aceea mergem doar până la `n // 2`.

4. Rotirea cu o poziție la stânga

Șirul `1 2 3 4 5` devine `2 3 4 5 1`: toți se mută o poziție spre stânga, iar primul ajunge la sfârșit. Ca la ștergere, păstrăm primul element undeva sigur, mutăm restul și îl punem la coadă.

```
v = [1, 2, 3, 4, 5]
n = len(v)
prim = v[0]
for i in range(n - 1):
    v[i] = v[i + 1]
v[n - 1] = prim
print(*v)
```

```
2 3 4 5 1
```

Rotirea la dreapta (`1 2 3 4 5` devine `5 1 2 3 4`) merge invers: păstrezi ultimul, muți elementele spre dreapta **de la sfârșit** și pui ultimul pe poziția 0.

```python
v = [1, 2, 3, 4, 5]
n = len(v)
ultim = v[n - 1]
for i in range(n - 1, 0, -1):
    v[i] = v[i - 1]
v[0] = ultim
print(*v)
```

```
5 1 2 3 4
```

5. Verifică-te

1. Ce valori au `a` și `b` după `a, b = 3, 7` urmat de `a, b = b, a`?
2. Ce devine `[8, 2, 5]` după `v[0], v[2] = v[2], v[0]`?
3. De câte ori se repetă bucla de inversare pentru `n = 8`? Dar pentru `n = 9`?
4. Care este perechea lui `v[1]` la inversarea unui șir cu `n = 6`?
5. Rotește la stânga cu o poziție șirul `4 5 6 7`.

Răspunsuri

1. `a = 7`, `b = 3`.
2. `[5, 2, 8]`.
3. De 4 ori (`8 // 2`); de 4 ori (`9 // 2`), iar elementul din mijloc rămâne pe loc.
4. `v[4]`, pentru că `n - 1 - 1 = 4`.
5. `5 6 7 4`.