INFORMATICĂ · PYTHON · CLASA A VII-A · STRUCTURA REPETITIVĂ · LECȚIA 11

Repetiția cu condiție: while

Durată: 50 min Nivel: gimnaziu · Python Înainte: lecțiile py7g-10

CE VEI ȘTI DUPĂ LECȚIE

- Scrii o buclă while cu cele trei părți: pornire, condiție, modificare.
- Alegi între for și while după cum știi sau nu de câte ori se repetă ceva.
- Eviți și oprești o buclă fără sfârșit.

1. Repetăm cât timp...

Uneori **nu știm** de câte ori trebuie să repetăm: *„Pune bani în pușculiță până ajungi la 100 de lei”*, *„citește numere până scrie cineva 0”*. Atunci folosim `while` (în română: **cât timp**). Bucla se repetă **cât timp o condiție este adevărată**.

```
i = 1
while i <= 5:
    print(i, end=" ")
    i = i + 1
print("gata")
```

```
1 2 3 4 5 gata
```

2. Cele trei părți ale unui while

Orice buclă `while` are trei lucruri, care trebuie să fie **toate** acolo:

Partea	În exemplul de mai sus	Ce se întâmplă dacă lipsește
1. pornirea (înainte de buclă)	`i = 1`	`NameError`: `i` nu există
2. condiția (în rândul `while`)	`i <= 5`	bucla nu pornește sau nu se oprește
3. modificarea (în corp)	`i = i + 1`	**bucla nu se mai oprește**

Python verifică condiția **înainte de fiecare repetare**. Dacă este falsă chiar de la început, corpul nu se execută **niciodată**.

ATENȚIE — GREȘELI FRECVENTE

Bucla infinită. Dacă uiți modificarea (`i = i + 1`), condiția `i <= 5` rămâne adevărată pentru totdeauna, iar programul afișează la nesfârșit. Nu te speria: în Thonny apasă butonul roșu **Stop**, iar în IDLE apasă **Ctrl + C**.

3. Exemplu: pușculița

Ai 10 lei și îți dublezi banii în fiecare zi. În câte zile ajungi la cel puțin 100 de lei?

```
bani = 10
zile = 0
while bani < 100:
    bani = bani * 2
    zile = zile + 1
print(zile, "zile, ai", bani, "lei")
```

```
4 zile, ai 160 lei
```

Urmărirea pe pași: `10 → 20 → 40 → 80 → 160`. La 160 condiția `bani < 100` devine falsă și bucla se oprește.

4. Citire până la 0

Aici folosim `while` pentru că nu știm câte numere vor veni. Cel care introduce datele scrie **0** când termină.

```
suma = 0
x = int(input())
while x != 0:
    suma = suma + x
    x = int(input())
print(suma)
```

Pentru intrările `5`, `3`, `8`, `0`:

```
16
```

Atenție la model: **citești o dată înainte de buclă**, ca să ai pe ce testa condiția, apoi **citești din nou la sfârșitul corpului**. Valoarea `0` nu se adună, pentru că bucla se oprește înainte.

5. Cifrele unui număr cu while

Din lecția py7g-04 știi că `n // 10` șterge ultima cifră. Dacă repeți asta cât timp mai rămân cifre (`n > 0`), poți „mânca” numărul cifră cu cifră:

```
n = int(input())
c = 0
while n > 0:
    c = c + 1
    n = n // 10
print(c)
```

Pentru intrarea `4872`:

```
4
```

Pas	`n` la început	`c` după pas
1	4872	1
2	487	2
3	48	3
4	4	4
	0 (stop)	

6. for sau while?

	`for`	`while`
Când?	știi **de câte ori** se repetă	repeți **până se întâmplă ceva**
Exemplu	„afișează numerele 1...n”	„citește până la 0”, „cifrele unui număr”
Modificarea lui `i`	o face singur `range`	o scrii **tu** în corp

7. Verifică-te

1. Ce afișează?

```
x = 20
while x > 5:
    print(x, end=" ")
    x = x - 6
```

2. De câte ori se execută corpul buclei `i = 10` ↵ `while i < 5:` ↵ `····i = i + 1`?
3. De ce este infinită bucla `n = 5` ↵ `while n > 0:` ↵ `····print(n)`?
4. Cum arată bucla care taie ultima cifră a lui `n` cât timp `n` este mai mare ca 0?

Răspunsuri

1. `20 14 8` (după 8 vine 2, iar 2 nu este mai mare ca 5).
2. De **niciodată**: condiția `10 < 5` este falsă încă de la început.
3. Valoarea lui `n` nu se schimbă niciodată, deci condiția rămâne adevărată. Lipsește `n = n - 1`.
4. `while n > 0:` ↵ `····n = n // 10`.

Probleme de lucru în clasă (pbinfo)

Le rezolvăm împreună în ședință, în ordinea de mai jos.

- [ ] `#51` **suma** — ușoară
  https://www.pbinfo.ro/probleme/51/suma

- [ ] #52 **sumapare1** — ușoară
  https://www.pbinfo.ro/probleme/52/sumapare1

Temă pentru acasă (pbinfo)

Trimite soluțiile pe pbinfo și notează punctajul obținut. Bifează ce ai terminat.

- [ ] #53 **cateimpare** — ușoară
  https://www.pbinfo.ro/probleme/53/cateimpare
- [ ] #1261 **catepare** — ușoară
  https://www.pbinfo.ro/probleme/1261/catepare