INFORMATICĂ · PYTHON · CLASA A VII-A · DATE NUMERICE · LECȚIA 3

Variabile și tipuri numerice

Durată: 50 min Nivel: gimnaziu · Python Înainte: lecțiile py7g-02

CE VEI ȘTI DUPĂ LECȚIE

- Creezi variabile cu atribuirea și le folosești în calcule.
- Alegi nume corecte pentru variabile.
- Deosebești numerele întregi (int) de cele zecimale (float).

1. Variabila: o cutie cu nume

Un calculator trebuie să **țină minte** valori: scorul dintr-un joc, banii din buzunar, vârsta ta. Pentru asta folosim **variabile**. O variabilă este ca o **cutie cu etichetă**: eticheta este numele, iar în cutie se află o valoare.

```
scor = 10
print(scor)
```

```
10
```

Semnul `=` se numește **atribuire** și se citește „pune în". Rândul `scor = 10` înseamnă: *pune în cutia numită scor valoarea 10*. **Nu** este egalul din matematică!

2. Valoarea se poate schimba

În cutie încape o singură valoare. Când pui alta, **cea veche dispare**.

```
scor = 10
scor = 25
print(scor)
```

```
25
```

Variabila poate apărea chiar în dreapta atribuirii. Python calculează **întâi** partea din dreapta, apoi pune rezultatul în cutie:

```
scor = 10
scor = scor + 5
print(scor)
```

```
15
```

`scor = scor + 5` înseamnă: „ia valoarea lui `scor`, adaugă 5 și pune rezultatul înapoi în `scor`". Prescurtat: `scor += 5` (la fel există `-=`, `*=`).

3. Reguli pentru nume

Regulă	Corect	Greșit
doar litere, cifre și `_`	`numar_mere`, `x2`	`numar mere`, `x-2`
nu începe cu cifră	`punct1`	`1punct`
litere mari ≠ litere mici	`Scor` și `scor` sunt variabile **diferite**	
fără diacritice	`varsta`, `inaltime`	`vârstă`

Alege nume care **spun ce conțin**: `pret` este mai bun decât `p`. Variabilele mici (`a`, `b`, `n`) sunt bune la probleme de matematică.

> **ATENȚIE — GREȘELI FRECVENTE**
>
> Nu poți folosi o variabilă înainte să-i dai o valoare! Programul `print(x)` dă `NameError: name 'x' is not defined`.

4. Variabile în calcule

```
latime = 4
lungime = 7
aria = latime * lungime
print("Aria este", aria)
```

```
Aria este 28
```

`print` poate afișa **mai multe lucruri** separate prin virgulă. Între ele pune automat câte un spațiu.

5. Tipuri de numere

Python deosebește două tipuri de numere:

Tip	Ce este	Exemple
int	număr **întreg**	`7`, `-3`, `0`, `1000`
float	număr **cu zecimale**	`2.5`, `-0.75`, `3.0`

În Python, **separatorul zecimal este punctul**, nu virgula: scrii `2.5`, nu `2,5`.

```
a = 7
b = 2.5
c = 3.0
print(type(a), type(b), type(c))
```

```
<class 'int'> <class 'float'> <class 'float'>
```

`type(...)` îți spune tipul valorii. Atenție: `3` și `3.0` au aceeași valoare, dar **tipuri diferite**. Când calculezi cu un `float`, rezultatul este tot `float`:

```
pret = 2.5
bucati = 4
print(pret * bucati)
```

```
10.0
```

DE ȘTIUT

Tipul unei variabile se schimbă odată cu valoarea. După `x = 5`, `x` este `int`; după `x = 5.5`, devine `float`.

6. Verifică-te

1. Ce afișează?

```
a = 4
a = a * 3
print(a)
```

2. Care nume sunt corecte: `nota_1`, `2nota`, `nota finala`, `Nota`?
3. Ce tip are `x` după `x = 8`? Dar după `x = 8.0`?
4. Ce afișează?

```
x = 5
y = x
x = 9
print(x, y)
```

Răspunsuri

1. `12`.
2. Corecte: `nota_1` și `Nota`. `2nota` începe cu cifră, `nota finala` are spațiu.
3. `int`, respectiv `float`.
4. `9 5`: `y` a primit valoarea 5 și nu mai este legat de `x`.