

## Limbajul Python

- Creat în 1991 de Guido van Rossum
- Permite
  - programare procedurală  $\nwarrow$  structuri de date /  $\Rightarrow$  prog.
  - programare orient. ob. (POO)  $\nwarrow$  algoritmi
  - programare funcțională  $\nwarrow$  struct. de date  $\nwarrow$  algoritmi  $\rightarrow$  act.

ex: progr. proced.

```
void sortare (int v[100], int m) { ... }
int a [100], m;
sortare (a, m);
```

ex: POO

```
class Vector {
    int v[100], m; // str. date
    public:
        void sortare();
}

int a [100], m;
Vector t(a, m); // construim un ob. t
t.sortare();
afisare t.v;
```

$\rightarrow$  arhitectură

## Tipuri de date

ex:  $x = 100$        $x = \text{"test"}$   
 print(x)  $\Rightarrow$  100      print(x)  $\Rightarrow$  test

! Un tip de date = clasă în Python

! O valoare = un obiect și o var = o referință spre un obiect

## Clase

- ① `NoneType` → `None` (ct., un fel de `void`)
- ② Tipuri numerice
  - `int` ( $\mathbb{Z}$ )
  - `float` ( $\approx$  `double` din C++)
  - `complex` ( $\mathbb{C}$ ,  $x = 1 + 2j$ )
- ③ `Bool` → `True` / `False`
- ④ Tipuri secvențiale / indexate
  - șiruri de caractere ( $x = \text{"text"}$ )
  - tupleuri ( $x = (1, 2, 3)$ )
  - liste ( $x = [1, 2, 3]$ )
- ⑤ Tipuri multime (val. distincte, fără duplicate)
  - `set` ( $x = \{1, 2, 2, 3\} \Rightarrow x = \{1, 2, 3\}$ )
  - `frozenset`
- ⑥ Dichiarare → `dict` ( $d = \{ \text{'Pop Iarn'} : 3.50, \text{'Iarn Mihai'} : 8.40 \}$ )  
`print(d['Pop Iarn'])`

## Scriserea datelor

ex: `x = 100`  
`y = 10`  
`print(x, y)` // 100 10  
`print(f"{x} + {y} = {x+y}")` // 100+10

## Citirea datelor

ex: `x = input("Dati x: ")` // 100 // scrierea datelor  
`y = input("y = ")` // 10  
`print(x, y)` // "100" "10"

ex:  $x = \text{int}(\text{input}("x=")) // 100$

↳ funcție de conversie din carac. în nr.

## Operatori

• aritate = nr. de operanzi

ex:  $a = 9 - 6$  aritate = 2  
 $y = -x$  aritate = 1

• prioritate

• asociativitate

① Operatori aritmetici:  $+$ ,  $-$ ,  $*$ ,  $/$  ( $: \in \mathbb{R}$ ),  $//$  ( $: \in \mathbb{Z}$ ),  $\%$ ,  
 $**$  (ridicare la putere)

② Relationali:  $<$ ,  $\leq$ ,  $>$ ,  $\geq$ ,  $==$ ,  $!=$   
 $is$ ,  $is\ not$  (testarea identității)  
 $in$ ,  $not\ in$  (testarea la apartenență)

③ Pe bity

④ Logici: AND, OR, NOT (val. logice:  $True \approx \neq \neq False$ )  
 $x\ and\ y = \begin{cases} x & \text{dacă } x \approx False \\ y & \text{dacă } x \approx True \end{cases}$   
ex:  $100\ and\ "test." = "test."$   
 $0.0\ and\ "test." = 0.0$

⑤ Conditional:  $expr_1\ if\ conditie\ else\ expr_2$   
ex:  $max, min = x\ if\ x > y\ else\ y$

• Prioritățile operatorilor (dintr-unu)

$() \rightarrow \begin{cases} \text{apel de funcție} \\ \text{indexare } [ ] \\ \text{metodă / dată membră} \end{cases} \rightarrow ** \rightarrow \begin{cases} \sim x\ (NOT\ x) \\ *x, -x\ (\text{schimbare, minus}) \end{cases} \rightarrow *, /, //, \% \rightarrow +, -\ (\text{binar}) \rightarrow >>, << \rightarrow \<, \> \ (\text{și pe bity}) \rightarrow \wedge \rightarrow \vee \rightarrow \text{relationali} \rightarrow NOT \rightarrow AND \rightarrow OR \rightarrow \text{condiționali}$

ex:  $if\ x ==\ not\ y \Rightarrow \text{eroare}$