

# Introducere și sintaxă

ROSEdu Py4School  
noiembrie 2013

Mihnea Dobrescu-Balaur

# Python

```
def apartine(x, v):  
    for nr in v:  
        if nr == x:  
            return True  
    return False
```

```
valori = [1, 2, 3 , 7, 13, -2, 4, 8, 9]  
print apartine(3, valori) # True  
print apartine(24, valori) # False
```

# Python

- limbaj dinamic
- nivel înalt
- expresiv
- ușor de înțeles

# De ce un limbaj dinamic?

- gestiunea memoriei se face automat
- mai puține erori de implementare
- pune accentul pe gândirea algoritmică

# De ce nivel înalt?

- abstractizează conceptele hardware (registri, memorie, pointeri etc.)
- funcții accesibile pentru acțiunile uzuale (fișiere, procesare de text, grafică, rețea)
- viteză rapidă de implementare - feedback constant

# Expresivitate

- sintaxa concisă ne permite să facem mai mult scriind mai puțin (fără a pierde din claritate)
- $3 < x < 10$
- `result = True if is_correct(answer) else False`
- `pare = [x for x in naturale if x % 2 == 0]`

# Claritate

- sintaxa este inspirată din pseudocod și din matematică
- se folosesc cuvinte (and în loc de &&) pentru a face limbajul mai ușor de citit
- indentarea face parte din sintaxă - codul altora devine ușor de citit

# Față de C/C++

- mai ușor de înțeles
- pune accentul pe algoritm și nu pe implementare
- testare facilă
- funcțiile existente alături de bibliotecile externe reduc timpul total de implementare



# Sintaxă

- nu specificăm tipuri
- indentare
- fără acolade
- cuvinte - and, or, True, False

# Sintaxă (cont.)

- nu avem do..while
- while ca în C
- for diferit - enumeră elementele unei colecții

# Tipuri de date

- numere
- liste (vectori)
- dicționare
- șiruri de caractere

