Python pentru aplicații
Modul 1 – Introducere în Python

2. Elemente de bază ale limbajului

Titus Adrian Beu
Signum Data SRL
Universitatea Babeș-Bolyai, Facultatea de Fizică

Conținutul capitolului

- **Moduri de lucru** – interactiv, cu scripturi
- **Identificatori, cuvinte cheie, instrucțiuni**
  - Comentarii
  - Identificatori Python, cuvinte cheie (cuvinte rezervate)
  - Instrucțiuni continuate, instrucțiuni multiple
  - Blocuri de instrucțiuni, indentare
- **Tipuri de date simple, variabile, conversii**
  - Tipuri numerice (int și float) și stringuri
  - Variabile locale și variabile globale
  - Conversia tipurilor de date
- **Operatori**
  - Operatori Python – aritmetici, pe biți, de atribuire, de comparare, logici, de identitate, de apartenență.

Moduri de lucru – interactiv, cu scripturi

Modul de lucru interactiv

- Deschidem un terminal în VS Code cu Terminal / New Terminal (Ctrl + `) și lansăm Python:

```
PS C:\Users\titus> python
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- La prompt >>> se pot introduce comenzi, care sunt executate cu Enter:

```
>>> print("Hello, Python!")
Hello, Python!
>>>
```

- Se poate tipări o expresie sau o variabilă prin simpla invocare (nu e necesar print):

```
>>> a = 1
>>> b = 2
>>> c = a + b
>>> c
3
>>>
```


Modul de lucru interactiv

- Deschidem un notebook Jupyter în VS Code cu New File / Jupyter Notebook
- Introducem codul in prima celulă și o executăm cu Ctrl + Enter

```
print("Hello, Python!")
```
```
Hello, Python!
```

- Creăm o nouă celulă de adnotare cu butonul +Markdown
- Introducem comentariul și executăm celula cu Ctrl + Enter:

```
Operatie aritmetica
```

- Creăm o nouă celulă de cod cu butonul +Code
- Introducem comentariul și executăm celula cu Ctrl + Enter:

```
a = 1
b = 2
a + b
```
```
3
```


Modul de lucru cu script-uri

- Creăm un script Hello.py cu IDLE sau VS Code:

```
print("Hello, Python!")
```

- Executăm script-ul din IDLE sau VS Code cu Ctrl + F5 și output-ul în terminal este

```
Hello, Python!
```

- Scriptul poate fi executat în File Explorer cu dublu click pe fișier, sau într-un terminal cu comanda

```
>>> python Hello.py
Hello, Python!
```

Identificatori, cuvinte cheie, instrucțiuni

Comentarii

- Comentariile nu influențează execuția – sunt total ignorate de interpretorul Python.
- Sunt esențiale pentru:
  - a documenta codul pentru alți utilizatori
  - a clarifica anumite secțiuni pentru alți programatori/dezvoltatori.
- Tipuri de comentarii:
  - pe o linie – încep cu caracterul # (hash)
  - pe mai multe linii (blocuri) – intercalate între perechi de trei caractere """
  - linii goale – spațiere vizuală

```
# Comentariu pe o linie

print("Hello, Python!")      # comentariu pe o linie executabila

"""
Bloc de comentarii
pe mai multe linii
"""
```


Identificatori Python

- **Identificatori** – nume utilizate pentru a identifica variabile, funcții, clase, module și alte obiecte.
- Un identificator se compune din unul sau mai multe caractere – exclusiv (A - Z, a - z, 0 - 9, _) – litere, cifre și caracterul de subliniere.
- Un identificator Python nu poate să înceapă cu o cifră și nu poate să conțină caractere speciale (~, !, @, #, $, %, ^, &, *, -, +, =, ...) și caracterul spațiu.
- Identificatori legali: fact1, ValMax, val_min, _amax
- Identificatori ilegali: 1fact, Val@Max, val-min, ~amax
- Python este sensibil la majuscule / minuscule – ValMax și valMax sunt identificatori diferiți.
- **Convenții pentru identificatorii Python**:
  - Numele claselor Python încep cu majuscule.
  - Toți ceilalți identificatori încep cu o literă mică.
  - Începerea cu un singur caracter de subliniere indică un identificator privat.
  - Începerea cu două caractere de subliniere indică un identificator puternic privat.
  - Încheierea cu două liniuțe de subliniere indică un nume special definit în limbaj.

Cuvinte cheie (cuvinte rezervate)

- **Cuvintele cheie** sunt elemente fundamentale unice ale oricărui limbaj de programare.
- Cuvintele cheie nu pot fi folosite ca identificatori – este emis un mesaj de eroare de sintaxă.

```
# Lista cuvintelor cheie Python

import keyword                                      # importa modulul keyword.py

print("Lista cuvintelor cheie Python\n")

print(keyword.kwlist)                          # tipareste lista cuvintelor cheie


Lista cuvintelor cheie Python

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class',
'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global',
'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise',
'return', 'try', 'while', 'with', 'yield']
```


Cuvinte cheie (cuvinte rezervate)

- **Cuvintele cheie** sunt elemente fundamentale unice ale oricărui limbaj de programare.
- Cuvintele cheie nu pot fi folosite ca identificatori – este emis un mesaj de eroare de sintaxă.

```
# Lista cuvintelor cheie Python

import keyword                                  # importa modulul keyword.py

help("keywords")                          # tipareste lista cuvintelor cheie


Here is a list of the Python keywords.  Enter any keyword to get more help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not
```


Instrucțiuni continuate

- **O instrucțiune** este o secțiune de cod care poate fi executată individual de interpretorul Python.
- O instrucțiune se reduce în mod tipic la o linie și se încheie cu caracterul ascuns \n (NEWLINE).
- În Python, caracterul ";" nu este folosit ca terminator de instrucțiuni!
- O instrucțiune poate fi continuată în linia următoare cu caracterul \ sau paranteze (...), [...], {...} (notebook):

```
                                # instructiune continuata pe mai multe linii cu \
a = 1 + \
    2 + \
    3
print("a =", a)

b = ( 1 +                                       # expresie continuata intre ()
      2 +
      3 )
print("b =", b)

c = [ 1 +                                       # expresie continuata intre []
      2 +
      3 ]
print("c =", c)

a = 6
b = 6
c = [6]
```