



Tema 1.

Introducere în Python



Ce ne așteaptă?

1. Noțiuni de Python

2. Instrucțiuni de codare

3. Identificatori în Python

1. De ce Python?
2. Ce este Python?
3. Ce versiuni de Python există?
4. Care sunt caracteristicile lui Python?
5. Unde este utilizat Python?
6. Care este procedura de instalare Python?
7. Care sunt modalitățile de a scrie un program Python?
8. Care este fluxul de execuție a programului Python?
9. Ce sunt identificatorii în Python?
10. Care sunt regulile pentru definirea identificatorilor?
11. Care sunt cuvinte cheie în Python?
12. Cum se realizează comentariile în Python?
13. Ce reprezintă indentarea în Python?

1. Noțiuni de Python

De ce Python ?

La ora actuală cel mai popular limbaj de programare

<https://www.tiobe.com/tiobe-index/>

<https://pypl.github.io/PYPL.html>

<https://spectrum.ieee.org/top-programming-languages/#toggle-gdpr>

https://www.youtube.com/watch?v=UNSoPa-XQN0&ab_channel=CleanCoders

Limbaje high-level vs limbaje low-level

Limbaje high-level - programmer-friendly

Exemple – Java, C, C++, Python

```
public class FirstProgram
{
    Public static void main(String args[])
    {
        System.out.println("Hello World");
    } //End of Main
} //End of class
```

Limbaje low-level - machine-friendly

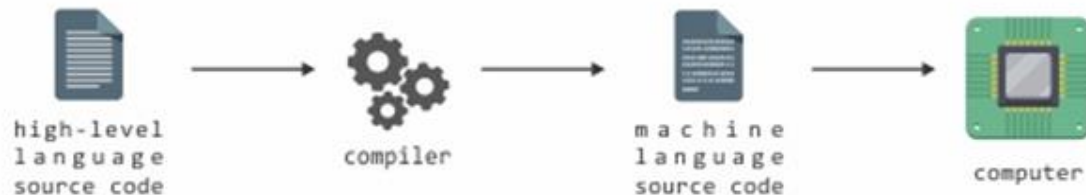
Exemple – limbajul Assembler, limbajul mașină

0736:010D 48	DEC	AX
0736:010E 65	SEG	GS (unused)
0736:010F 6C	INSB	
0736:0110 6C	INSB	
0736:0111 6F	OUTSW	
0736:0112 2124	AND	[SI],SP

Compiler vs Interpretator

Compilerul transformă întregul cod sursă în cod mașină executabil în cadrul unei singure operații.

Exemple – Java, C++



Interpretatorul transformă linie cu linie codul sursă în cod mașină executabil.

Exemple – PHP, Ruby



Definiția lui Python

● **Python este un limbajul de programare:**

- cu scop general
- high-level
- atât interpretator cât și compilator

Versiuni Python

1. Python 1.0V introdusă în ian. 1994
2. Python 2.0V introdusă în oct. 2000
3. Python 3.0V introdusă în dec. 2008

Caracteristicile Python

- **Simplitate** (vezi Hello, World!)
- **Open source**
- **Tipizare dinamică**
- **Independent de platforma**
- **Portabil**
- **Librarii imense**
- **Conectivitate la baza de date**

Utilizare Python

- **Aplicații independente (Standalone applications)**
- **Aplicații web**
- **Data science**
- **Machine learning și inteligență artificială**
- **Conexiuni la baze de date**
- **Automatizări**
- **Securitatea datelor**

2. Instrucțiuni de codare

Instalarea Python

- Descărcare Python:

<https://www.python.org/downloads/>

- Instalare Python:

https://utm-my.sharepoint.com/:f:/g/personal/pavel_nicolaev_tlc_utm_md/EgDbVwX6JPZCjQzHeuXbmE4BocPnM_3fyVj6KGhcMc0seA?e=TXQcIE

- bifare “Add Python 3.X to PATH”

Modalități de scriere cod Python

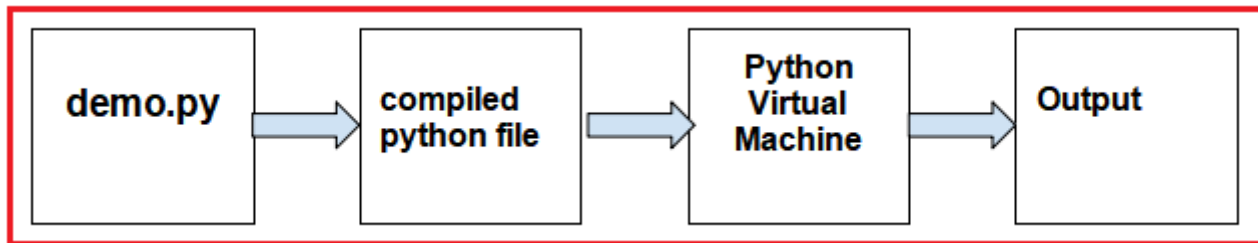
- **Direct în cmd după activarea comenzii python**
- **Utilizând un editor de text și executarea în cmd**
 - Notepad, Sublime Text, Visual Studio Code și altele
- **Utilizând un IDE**
 - PyCharm, Jupyter Notebook, VS Code, Atom și alte
- **Utilizând medii online**

<https://replit.com/>

<https://glot.io/>

Un fișier Python are extensia ‘.py’ sau ‘.python’

Fluxul de executare a programului



- **comanda cmd:** `python -m py_compile demo.py`

- **directoriu:** “`__pycache__`”

3. Identificatori

Identificatori în Python

1. nume clasă
2. nume pachet
3. nume variabilă
4. nume modul
5. nume funcție
6. nume metodă

Reguli de definire a identificatorilor

1. pot conține litere ale alfabetului – majusculele și minusculele sunt tratate diferit.
2. pot conține cifre (0 la 9) dar nu pot începe cu o cifra
3. pot conține simbolul Underscore (_)
4. nu pot reprezenta un cuvânt cheie
5. nu pot conține alte simboluri (\$,!,#, @)

- Syntax Error

- **Cuvintele cheie:**

'False', 'None', 'True', 'and', 'as', 'assert', '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'

Exemple de identificatori

1. `435student` → **invalid** (nu trebuie să înceapă cu o cifră)
2. `student564` → valid
3. `student565info` → valid
4. `$student` → **invalid** (nu trebuie să conțină simboluri speciale)
5. `_student_info` → valid (poate să înceapă cu underscore)
6. `class` → **invalid** (nu poate fi un cuvânt cheie)
7. `def` → **invalid** (nu poate fi un cuvânt cheie)

Practici bune în definirea identificatorilor

1. clasă	<ul style="list-style-type: none">- Prima literă va fi o majuscula- Dacă e compusa din mai multe cuvinte fiecare va incepe cu o majusculă- Exemplu: StudentInfo
2. Pachete	<ul style="list-style-type: none">- Toate literele vor fi minuscule- Dacă e compusa din mai multe cuvinte se vor diviza prin simbolul underscore (_)- Exemplu: numele_meu
3. Module	
4. Variabile	
5. Funcții	
6. Metode	
7. Constante	<ul style="list-style-type: none">- Toate literele vor fi majuscule- Dacă e compusa din mai multe cuvinte se vor diviza prin simbolul underscore (_)- Exemplu: NUMELE_MEU
8. Variabile private	<ul style="list-style-type: none">- Variabilele vor începe cu simbolul underscore (_)- Exemplu: _nume
9. Entități non-accesibile	<ul style="list-style-type: none">- Aceste entități vor fi încadrate în simboluri underscore (_) duble- Exemplu: __init__

Practici bune în definirea identificatorilor

- **Să fie descriptivi**

`a = 10 #valid dar nerecomandat`

`nr_student= 10 # valid și recomandat`

- **Lipsa abrevierilor inutile**

`nr_student_banca_4_mijloc = 10 #valid dar nerecomandat`

`nr_student = 10 #valid și recomandat`

Comentarii în Python

- Comentariile reprezintă cea mai simplă cale de descriere a codului
- Python ignoră comentariile în procesul de execuție a programului .
- Pentru a comenta un cod pe o linie se adaugă simbolul hash #
- Pentru comentariile multi-linie se utilizează 3 semne apostrof (''' ''') sau 3 semne ghilimele (""" """)
- Exemple:

```
# Comentariu pe o linie
```

```
'''Comentariu pe mai  
multe linii'''
```

Indentarea în Python

- **Este obligatorie în cazul grupării declarațiilor**
 - Indentarea menține separat un grup de declarații
 - Se recomandă ca indentarea să fie realizată prin 4 spații sau un tab
 - Trebuie respectată ordinea indentării în caz contrar apar erori `IndentationError`