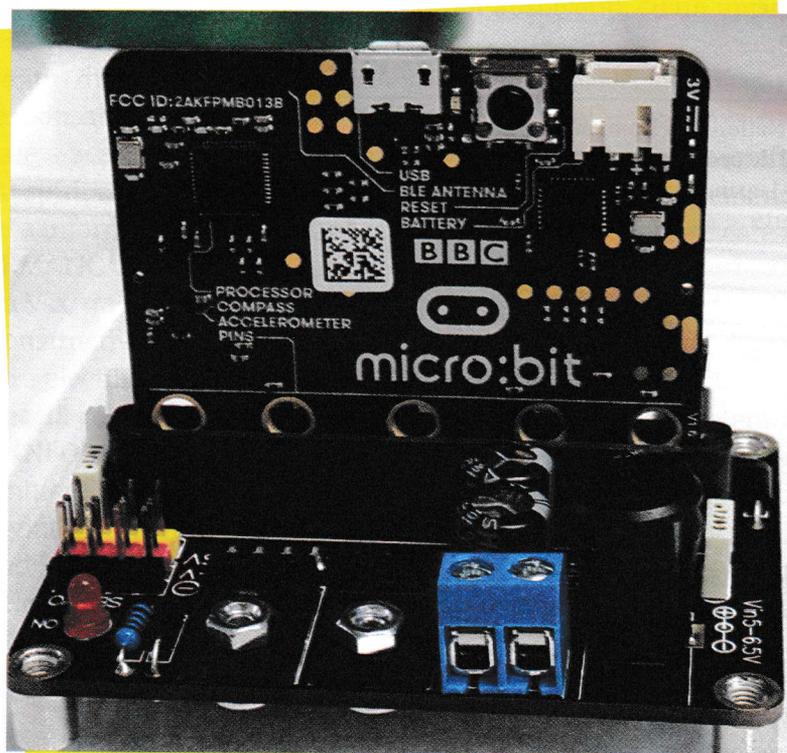


INFORMATICĂ ȘI TIC

- Programare în limbajul **Python**

- BBC Micro Bit **Micro:bit**



Prefață.....	3
--------------	---

Limbajul Python

Limbajul Python.....	6
Mediul limbajului de programare Python.....	6
Elemente de bază ale limbajului Python.....	12
Structura unui program în limbajul Python.....	14
Date în limbajul Python.....	14
A. Tipuri de date standard.....	14
B. Constante și variabile.....	15
C. Expresii.....	17
Operații de citire și afișare/scriere a datelor în Python.....	24
Instrucțiuni/comenzi pentru implementarea structurilor de control în limbajul Python.....	28
Modulul MATH.....	38
Șiruri de caractere în Python – String-uri.....	38
Liste în Python.....	43
Jocuri în Python.....	47
Ghicește numărul (Guess number).....	47
Urmărește (Follow me).....	48
Probleme rezolvate.....	51

BBC Micro Bit (micro:bit)

BBC Micro Bit (micro:bit).....	70
Interfața.....	71
Afișarea unui număr.....	71
Afișarea șirurilor de caractere.....	72
Afișarea unor simboluri predefinite (icon).....	73
Afișarea unor caractere defninite de utilizator.....	73
Butoanele lui micro:bit.....	74
Senzorul de temperatură.....	74
Busola (Compass).....	75
Luminozitate.....	76

LIMBAJUL PYTHON

Respect pentru oameni și cărți

Python este un limbaj interpretat, de scripting. El este ușor de utilizat și este independent de sistemul de operare utilizat. Aplicațiile Python sunt foarte ușor de depanat, codul putând fi cu ușurință inspectat în timpul rulării. Extensia programelor scrise în Python este **.py**.

Știați că...



Limbajul Python a fost dezvoltat în 1991 de către programatorul olandez Guido van Rossum. Versiunea Python 2, lansată în 2000, conține câteva elemente noi și permite dezvoltarea limbajului de o întreagă comunitate de programatori. Python 3 a fost lansat în 2008 și nu este compatibil cu Python 2. "Sfârșitul" lui Python 2 este prevăzut pentru anul 2020.

Fiind un limbaj dinamic, Python permite scrierea unei mari varietăți de programe, cum ar fi: programe specializate pentru procesare de texte, motoare de căutare, aplicații *web*. Python este un limbaj actual folosit de marile companii din domeniul IT (Google, Yahoo!, Amazon etc).

MEDIUL LIMBAJULUI DE PROGRAMARE PYTHON

A. INSTALARE PYTHON

Pentru început, vom instala pe calculator limbajul Python, de la adresa: www.python.org. De asemenea vom instala **Wing IDE** (ediția gratuită **Wing Personal**) de la wingware.com

Știați că...



Un mediu integrat de dezvoltare – **IDE**, este o aplicație software care oferă programatorilor mai multe facilități pentru dezvoltarea de software: scrierea, modificarea, compilarea, depanarea programului.

B. MODALITĂȚI DE LUCRU

B.1. Lucrul în IDLE Python.

După instalare, Python se poate lansa scriind **IDLE** la opțiunea de căutare a interfeței sistemului de operare.

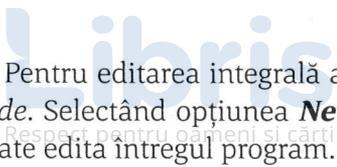
După lansare, se deschide *ferestra de comenzi* – *Shell* (care interpretează comenzile succesiv) din imaginea alăturată. Fiecare comandă scrisă în *Shell*, este verificată imediat și, dacă nu sunt erori, executată. Astfel, programul este verificat și executat comandă cu comandă.

Interfața ferestrei *Shell* permite, prin intermediul meniului, salvarea, editarea unui program, precum și setări ale interfeței.

IDLE (mediu integrat de dezvoltare și învățare) este un mediu pentru Python, instalat împreună cu limbajul.

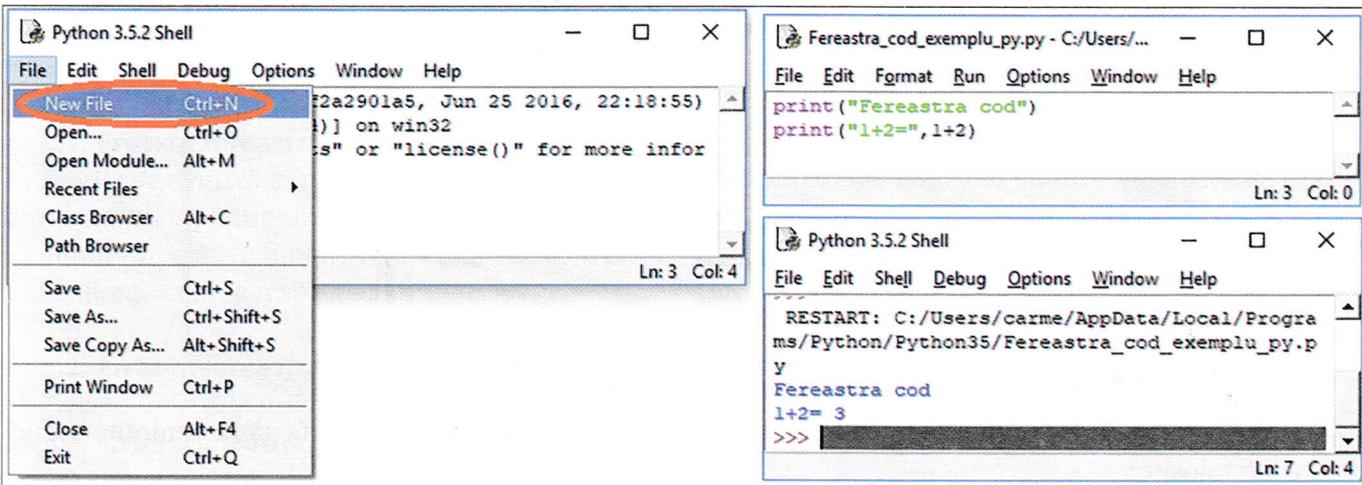
```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55)
[MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more infor
mation.
>>> |
```

Ln: 3 Col: 4



Pentru editarea integrală a unui program, Python permite deschiderea unei ferestre, denumită *ferastră Code*. Selectând opțiunea **New File** din meniul **File** al ferestrei *Shell*, se deschide fereastra *Code*, în care se poate edita întregul program. Interpretarea programului scris se va face la final, iar rezultatul executării programului va fi afișat în fereastra *Shell*. Interfața ferestrei *Code* permite ca programul scris să poată fi:

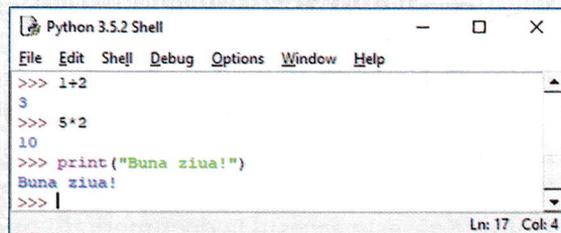
- salvat – opțiunea **Save** din meniul **File** (Ctrl+S);
- verificat – opțiunea **Check Module** din meniul **Run** (Alt+X);
- lansat în execuție, prin selectarea opțiunii **Run Module** din meniul **Run** (F5).



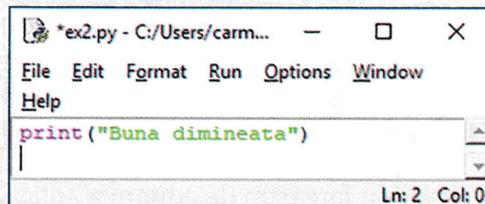
Exersați !

1.1. Tastați în fereastra *Shell*, după cursorul reprezentat prin ">>>", o operație aritmetică oarecare și apoi apăsați tasta **Enter**. Veți obține imediat rezultatul operației, fără a utiliza opțiuni din meniul interfeței.

1.2. Pentru a afișa un text oarecare, de exemplu "Buna ziua!", scrieți respectivul text între ghilimele sau apostrofuri, între cele două paranteze ale funcției **print**. Apăsați tasta **Enter**.



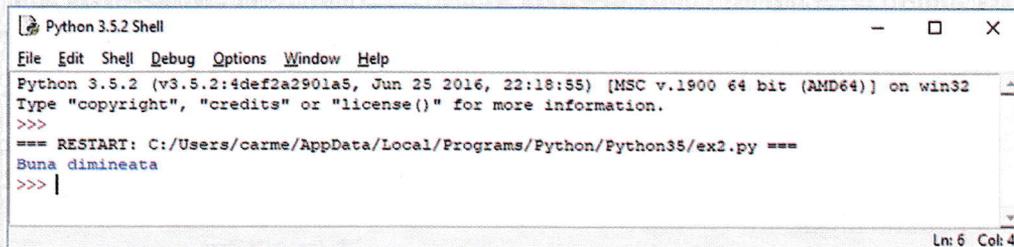
2.1. Scrieți textul de la exercițiul anterior în *ferestra Code*. Nu uitați să încadrați textul între ghilimele sau apostrofuri, între cele două paranteze ale funcției **print**.



2.2. Selectați opțiunea **Save** din meniul **File** al ferestrei *Code*. Introduceți numele fișierului, *Ex2.py*, apoi apăsați butonul **Save**.

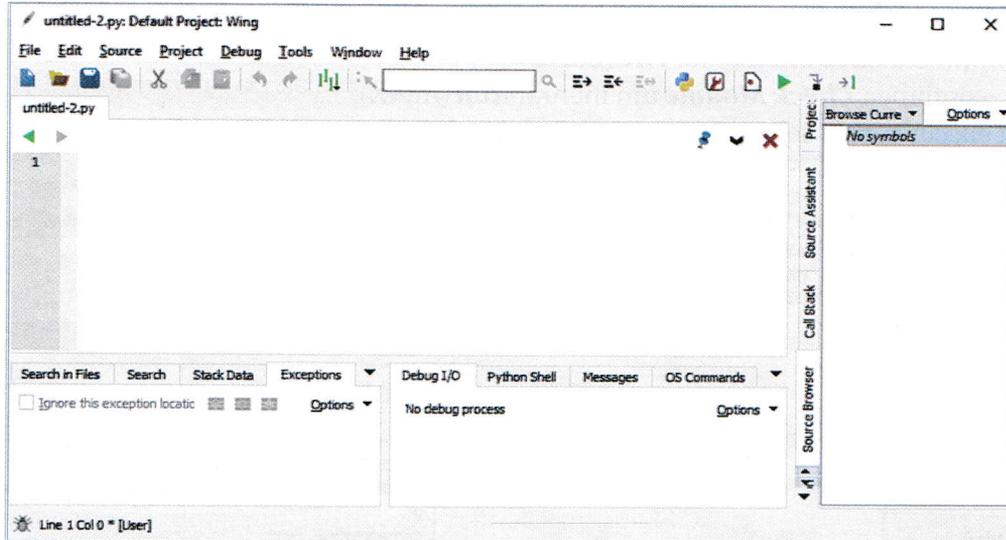
2.3. Verificați existența erorilor, selectând opțiunea **Check Module** din meniul **Run**. Dacă sunt semnalate erori, corectați-le.

2.4. Lansați în execuție programul, selectând opțiunea **Run Module** din meniul **Run**. În fereastra *Shell* vor fi afișate rezultatele executării programului.



B.2. Lucrul în Wing IDE Personal

Apăsați pe butonul **START** al interfeței sistemului de operare și dați clic pe **Wing IDE Personal**. Se lansează aplicația cu interfața de mai jos:

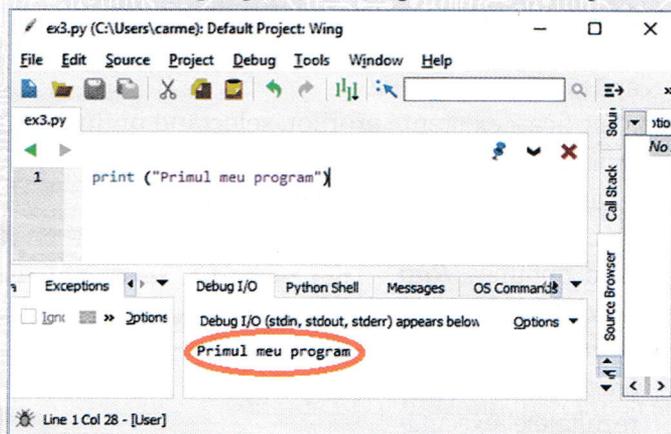


Amintim câteva dintre opțiunile uzuale din meniul interfeței, urmând ca celelalte să fie amintite atunci când vor fi folosite:

- File:**
 - New** (crearea unui nou fișier);
 - Open** (deschiderea unui fișier existent);
 - Save, Save as** (salvarea unui fișier Python – reamintim extensia py);
 - Quit** (ieșirea din aplicație);
- Edit:**
 - Undo**, **Redo**, **Line Numbers** (numerotarea liniilor programului);
 - Cut**, **Copy**, **Paste**, **Preferences** (particularizarea interfeței Wing);
- Debug:** opțiuni pentru depanarea programului / lansare în execuție (F5);
- Tools:** unelte/ferestre necesare programatorului;
- Help:** se găsește accesul la documentații privind *Wing*, cât și *Python*.

Exersați !

1. Scrieți în fereastra de editare a aplicației *Wing* textul "*Primul meu program*", între ghilimele sau apostrofuli, între cele două paranteze ale funcției `print`.
2. Selectați opțiunea **Save** din meniul **File**. Introduceți numele fișierului, *Ex3.py*, iar apoi apăsați butonul **Save**.
3. Lansați în execuție programul, selectând opțiunea **Start/Continue** din meniul **Debug** sau apăsați . Dacă apare vreun mesaj de eroare în colțul din stânga jos a ferestrei, scrisă cu roșu, corectați-o și relansați în execuție. Vizualizați rezultatul afișat.



C. DEPANAREA PROGRAMELOR PYTHON

- Operația de corectare a erorilor depistate la un program se numește depanare. Cele mai frecvente erori sunt erorile de sintaxă (*Syntax Error*), de indentare și cele de nume (*Name Error*).
- Erorile de sintaxă, de regulă, sunt generate de greșeli de scriere, ortografie sau de o spațiere incorectă în cod care împiedică executarea liniei (indentare incorectă).
- Apelul unor funcții care nu există va genera un mesaj de eroare de nume, *NameError*, când se încearcă executarea liniei respective.
- Codul este executat linie cu linie. Întâlnirea unei erori de sintaxă întrerupe programul, fiind necesară remedierea erorii.

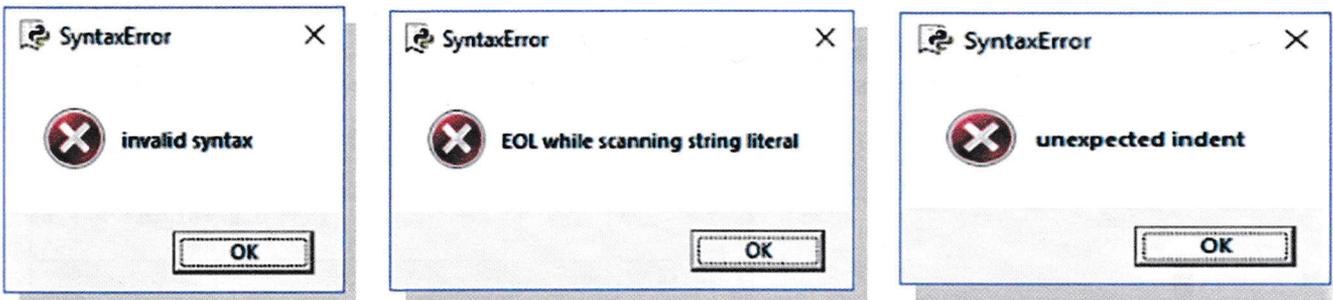
C.1. Erori în fereastra Shell

Erorile de sintaxă sunt raportate la finalul fiecărei linii, dacă există, marcând cu roșu linia curentă ce conține eroarea. Apariția unei linii incorecte va duce la întreruperea executării programului. Astfel, dacă – de exemplu – apelăm o funcție care nu există, vom primi un mesaj de eroare de nume, *NameError*, când interpretorul ajunge la linia respectivă.

C.2. Erori în fereastra Code

Fiecare linie este interpretată și executată succesiv, ca în fereastra *Shell*. Dacă apare o eroare la o linie, de regula, apare o fereastra pop-up cu un mesaj de eroare. Aceasta întrerupe executarea programului și este necesară corectarea lor.

Erorile de sintaxă sunt primele anunțate prin apariția unei ferestre cu mesajul *SyntaxError*.



Erorile de tip *NameError* apar după corectarea erorilor de sintaxă și sunt specificate în fereastra *Shell*.

```

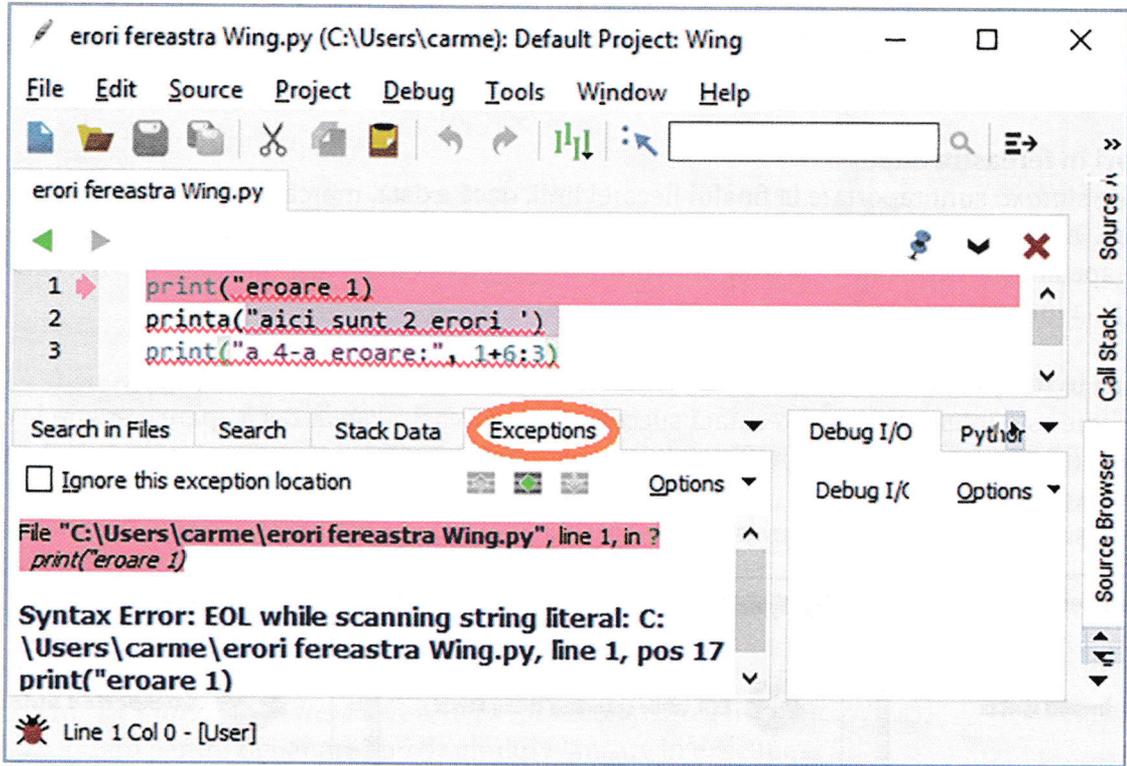
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
>>>
RESTART: C:/Users/carme/AppData/Local/Programs/Python/Python35/erori_fereastră cod.py
eroare 1
Traceback (most recent call last):
  File "C:/Users/carme/AppData/Local/Programs/Python/Python35/erori_fereastră cod.py",
line 2, in <module>
    printa("aici sunt 2 erori ")
NameError: name 'printa' is not defined
>>> |
Ln: 13 Col: 4

```

C.3. Erori în aplicația Wing

Dacă programul are erori, atunci prima linie ce conține eroare va fi subliniată cu roșu.

În fereastra corespunzătoare *Debug Process Exceptions* este precizat tipul erorii și semnificația ei. Se corectează eroarea și se selectează **Restart Debugging** din meniul **Debug**, pentru a vedea linia și semnificația următoarei erori, ș.a.m.d până la corectarea programului. Erorile de sintaxă sunt primele indicate, apoi erorile de nume, dacă există în program.



Atenție !

Sfaturi pentru depanarea celor mai frecvente erori:

- Ați verificat dacă este scris corect tot codul?
- Ați fost atenți la literele mari și cele mici? Python este un limbaj case-sensitiv!
- Ați folosit o pereche de ghilimele/caractere apostrof pentru afișarea mesajelor?
- Ați confundat cumva semnul minus (-) cu liniuța de subliniere (_) ?
- Ați utilizat parantezele corect? Parantezele/acoladele precum (), [] și { } sunt folosite în scopuri diferite. Verificați dacă sunt corecte parantezele utilizate și dacă sunt de același tip.
- Ați verificat dacă există spații în plus sau în minus la începutul liniei?
- Ați verificat liniile de deasupra și dedesubtul liniei evidențiate?
- Folosiți versiunea corectă de Python? Nu toate programele scrise în Python 3 sunt corecte în Python 2.

Se consideră programul *ex4.py* din imaginea alăturată. El conține erori.

Scrieți programul în cele trei moduri:

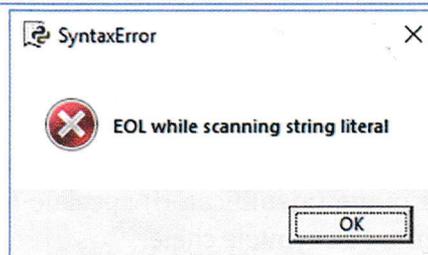
a) în fereastra *Shell*; b) în fereastra *Code*; c) în aplicația *Wing*.

Vizualizați și corectați cele 4 erori.

```
ex4.py
print ("eroare 1)
printa ("aici sunt 2 erori')
print ("a 4-a eroare:", 1+6:3)
```

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
>>> print("eroare 1)
SyntaxError: EOL while scanning string literal
>>> print("eroare 1")
eroare 1
>>> printa("aici sunt 2 erori ')
SyntaxError: EOL while scanning string literal
>>> printa("aici sunt 2 erori ")
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    printa("aici sunt 2 erori ")
NameError: name 'printa' is not defined
>>> print("aici sunt 2 erori ')
SyntaxError: EOL while scanning string literal
>>> print("aici sunt 2 erori ")
aici sunt 2 erori
>>> print("a 4-a eroare:", 1+6:3)
SyntaxError: invalid syntax
>>> print("a 4-a eroare:", 1+6/3)
a 4-a eroare: 3.0
>>>
```

```
ex4.py - C:/Users/carme/A...
File Edit Format Run Options Window Help
print("eroare 1)
printa("aici sunt 2 erori ')
print("a 4-a eroare:", 1+6:3)
|
Ln: 4 Col: 0
```



Știați că... ?

- Un compilator este un program ce transformă în cod binar codul sursă scris de programator.
- Un interpretor execută codul sursă direct, comandă după comandă, compilând comenzile în binar sau *bytecode*. *Bytecode* este un cod intermediar care se execută de o mașină virtuală asociată cu interpretorul. Python este un limbaj interpretat prin *bytecode*. Astfel, un program Python va fi executat doar pe un computer ce dispune de un interpretor.
- *Python* este un limbaj de scripting.

Mini proiecte

Căutați informații în mediul online și realizați o prezentare (folosind instrumentele TIC pe care le-ați studiat) pentru o temă aleasă dintre următoarele: a) compilatoare și interpretoare; b) limbaje de scripting.