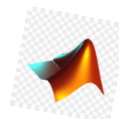


Lucrare individuala la Grafica pe calculator

TI-211

Popa Catalin



Laborator nr.1

Ex.1_1

a) Care va fi răspunsul obținut după rularea codului în fereastra de program IDLE:

```
y =20
y = 15
y = 30
print(y)
print(type(y))
```

Raspuns:

```
30
<class 'int'>
```

b) Care va fi răspunsul obținut după rularea codului în fereastra de program IDLE:

```
y =20
y = 15
y = 30
#print(y)
#print(type(y))
x = 5
z = y / x
print (z)
print(type(z))
```

Raspuns:

```
6.0
<class 'float'>
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Ex.1_2

a) Care va fi răspunsul obținut după rularea codului în fereastra de program IDLE:

```
print("Vasile", "are", 30, "de ani")
#print("Vasile", "are", 30, "de ani", end = " ... ")
#print ("Vasile", "Ion", "Ana", sep = " || ", end = " : Cine e fata?")
#print ()
#print ("Fata este Ana!!!")
```

Raspuns:

Vasile are 30 de ani

b) Care va fi răspunsul obținut după rularea codului în fereastra de program IDLE:

```
print("Vasile", "are", 30, "de ani")
print("Vasile", "are", 30, "de ani", end = " ... ")
#print("Vasile", "are", 30, "de ani", end = " ... ")
#print ("Vasile", "Ion", "Ana", sep = " || ", end = " : Cine e fata?")
#print ()
#print ("Fata este Ana!!!")
```

Raspuns:

Vasile are 30 de ani
Vasile are 30 de ani ...

c) Care va fi răspunsul obținut după rularea codului în fereastra de program IDLE:

```
print("Vasile", "are", 30, "de ani")
print("Vasile", "are", 30, "de ani", end = " ... ")
print("Vasile", "Ion", "Ana", sep = " || ", end = " : Cine e fata? ")
#print ()
#print ("Fata este Ana!!!")
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Raspuns:

```
Vasile are 30 de ani  
Vasile are 30 de ani ... Vasile || Ion || Ana : Cine e fata?
```

d) Care va fi răspunsul obținut după rularea codului în fereastra de program IDLE:

```
print("Vasile", "are", 30, "de ani")  
print("Vasile", "are", 30, "de ani", end = " ... ")  
print("Vasile", "Ion", "Ana", sep = " || ", end = " : Cine e fata? ")  
print ()  
#print ("Fata este Ana!!!")
```

Raspuns:

```
Vasile are 30 de ani  
Vasile are 30 de ani ... Vasile || Ion || Ana : Cine e fata?
```

e) Care va fi răspunsul obținut după rularea codului în fereastra de program IDLE:

```
print("Vasile", "are", 30, "de ani")  
print("Vasile", "are", 30, "de ani", end = " ... ")  
print("Vasile", "Ion", "Ana", sep = " || ", end = " : Cine e fata? ")  
print ()  
print ("Fata este Ana!!!")
```

Raspuns:

```
Vasile are 30 de ani  
Vasile are 30 de ani ... Vasile || Ion || Ana : Cine e fata?  
Fata este Ana!!!
```

Ex.1_3

Scrieți codul pentru a determina aria și perimetrul unui dreptunghi dacă știm că lungimea este $a = 10$ cm, iar lățimea $b = 4$ cm. Aria: $A = a \times b$ iar Perimetrul: $P = 2(a + b)$.

```
a = 10
b = 4
p = 2*(a*b)
ar = a * b
print("\nPerimetrul = {:.2f} (cm)".format(p))
print("\nAria = {:.2f} (cm)".format(ar))
```

Raspuns:

```
Perimetrul = 80.00 (cm)
Aria = 40.00 (cm)
```

Ex.1_4

Care va fi răspunsul obținut după rularea codului în fereastra de program IDLE:

```
x = 25
y = 20
z = 3
f = 2

a = (x - y) * z / f
b = ((x-y) * z) / f
c = (x - y) * (z / f)

print("a = ", a)
print("b = ", b)
print("c = ", c)
```

Raspuns:

```
a = 7.5  
b = 7.5  
c = 7.5
```

Ex.1_5

Scrieți un cod în care să verificați dacă un număr este par sau impar.

```
num = int(input("Introduceti un numar: "))  
if (num % 2) == 0:  
    print("{0} este par".format(num))  
else:  
    print("{0} este impar".format(num))
```

Raspuns:

```
Introduceti un numar: 12  
12 este par  
  
Introduceti un numar: 13  
13 este impar
```

Ex.1_6

Scrieți un cod în care în primă fază să ridicați un număr la pătră (exponentul 2) după care să extrageți rădăcina pătrată din el ca să obțineți la răspuns numărul inițial.

```
import math  
num = int(input("Introduceti un numar: "))  
x = num ** 2  
y = math.sqrt(x)  
print("\nNumarul la patrat este =",x)  
print("\nRadacina patrata din el = ",y)
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Raspuns:

```
Introduceti un numar: 4  
Numarul la patrat este = 16  
Radacina patrata din el = 4.0
```

Ex.1_7

Scrieți codul pentru a determina Aria: $A = 6a^2$ și Volumul: $V=a^3$ unui cub cu latura $a=20$ cm.

```
import math  
a = 20  
x = 6 * pow(a, 2)  
y = pow(a, 3)  
print("\nAria = {:.2f} (cm)".format(x))  
print("\nVolumul = {:.2f} (cm)".format(y))
```

Raspuns:

```
Aria = 2400.00 (cm)  
Volumul = 8000.00 (cm)
```

Ex.1_8

Scrieți un cod în 3 moduri diferite care să ne dea rezultatul când baza este 4 și exponentul este 3, $4^3=?$

1)

```
import math  
print("Calculam puterea unui numar folosind o bucla while")  
baza = 4  
exponent = 3  
result = 1  
while exponent != 0:  
    result *= baza  
    exponent-=1  
print("Raspuns = " + str(result))
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Raspuns:

```
Calculăm puterea unui număr folosind o buclă while  
Raspuns = 64
```

2)

```
import math  
print("Calculam puterea unui numar folosind o bucla for")  
baza = 4  
exponent = 3  
result = 1  
for exponent in range(exponent, 0, -1):  
    result *= baza  
print("Raspuns = " + str(result))
```

Raspuns:

```
Calculam puterea unui numar folosind o bucla for  
Raspuns = 64
```

3)

```
import math  
print("Calculam puterea unui numar folosind functia pow().")  
baza = 4  
exponent = 3  
result = pow(baza, exponent)  
print("Raspuns = " + str(result))
```

Raspuns:

```
Calculam puterea unui numar folosind functia pow().  
Raspuns = 64
```


Ex.1_9

Scrieți un cod care să rezolve următoarea ecuație: $z = |x - y| * (x + y)$ dacă știm că $x = 4$, iar $y = 6$.

```
import math
x, y = 4, 6
r = abs(x - y) * (x + y)
print("Raspuns = {}".format(r))
```

Raspuns:

Raspuns = 20

Ex.1_10

Numele variabilelor. În acest exercițiu vom învăța să dăm nume corecte variabilelor.

Reguli de bază pentru numele variabilelor în *Python*:

- numele unei variabile poate să înceapă cu literă sau simbolul „_”. Exemple: continent_ sau _continent

- numele unei variabile poate să conțină litere, cifre și simbolul „_”. Exemple: martie_13 sau martie13

- numele unei variabile trebuie să conțină aceleași litere pe parcursul întregului cod:

Martie2020 și martie2020 vor fi două variabile diferite în *Python*

- nu este permis de folosit „cuvinte-cheie” în numele unei variabile. Exemple de „cuvinte-cheie”:

False, def, if, raise, None, del, import, return, True, elif, in, try, and, else, is, while, as,

except, lambda, with, assert, finally, nonlocal, yield, break, for, not.

Analizați codul de mai jos, copiați-l în fereastra de program IDLE și încercați să rulați acest cod. O să apară multe erori, sarcina dumneavoastră este să eliminați din

GC-Popa Catalin				Coal
M0	Coal	Semnat	Data	

cod și din funcția print () toate variabilele care sunt greșite astfel încât în final codul să poată fi rulat.

```
salariu = 20 # $ pe ora
```

```
Salariu = 22
```

```
salariu_ = 17
```

```
sALARIU = 14
```

```
salariu2 = 8
```

```
_SALARIU_ = 27
```

```
print(salariu, Salariu, salariu_, sALARIU, salariu2, _SALARIU_)
```

Raspuns:

20 22 17 14 8 27

Ex.1_11

Scrieți un cod care să transforme gradele Fahrenheit în grade Celsius și invers. Formula de transformare este: $F = 9/5 * C + 32$

```
import math
```

```
r = float(input("Introduceti temperatura: "))
```

```
n = int(input("Introduceti 1,daca transformati din Celsius sau 2,daca  
transformati din fahrenheit : "))
```

```
if (n == 1):
```

```
    f = (r * 9/5) + 32
```

```
    print('%.2f Celsius este: %.2f Fahrenheit' %(r, f))
```

```
elif (n == 2):
```

```
    f = (r - 32) * 5/9
```

```
    print('%.2f Fahrenheit este: %.2f Celsius' %(r, f))
```

				GC-Popa Catalin		Coal
M0	Coal		Semnat	Data		

Raspuns:

```
Introduceti temperatura: 37
Introduceti 1,daca transformati din Celsius sau 2,daca transformati din fahrenheit : 1
37.00 Celsius este: 98.60 Fahrenheit
```

```
Introduceti temperatura: 99
Introduceti 1,daca transformati din Celsius sau 2,daca transformati din fahrenheit : 2
99.00 Fahrenheit este: 37.22 Celsius
```

Ex.1_12

În ultimul exercițiu trebuie să transformați MPH (mile pe oră) în KMH (kilometri pe oră) și invers.

Formulele de calcul vor fi: $\text{MPH} = 0.6214 * \text{KMH}$ și $\text{KMH} = 1.6093 * \text{MPH}$

```
import math

r =float(input("Introduceti viteza: "))

n = int(input("Introduceti 1,daca transformati din MPH sau 2,daca
transformati din KMH : "))

if (n == 1):
    f = 1.6093 * r
    print('%.2f Mile pe ora este: %0.2f kilometri pe ora' %(r, f))
elif (n == 2):
    f = 0.6214 * r
    print('%.2f kilometri pe ora este: %0.2f mile pe ora' %(r, f))
```

Raspuns:

```
Introduceti viteza: 2
Introduceti 1,daca transformati din MPH sau 2,daca transformati din KMH : 1
2.00 Mile pe ora este: 3.22 kilometri pe ora
```

```
Introduceti viteza: 10
Introduceti 1,daca transformati din MPH sau 2,daca transformati din KMH : 2
10.00 kilometri pe ora este: 6.21 mile pe ora
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Laborator nr.2

Ex2_1

Acest exercițiu l-am rezolvat și în capitolul 1 la Ex1_12 când am transformat MPH în KMH. Acum ne propunem același lucru doar că valoare care vrem s-o transformăm să fie introdusă de utilizator, prin urmare în codul nostru trebuie să fie prezentă și funcția `input ()`. Scrieți codul astfel încât răspunsul obținut să arate în felul următor:

```
Care este viteza în km/h?: 30
Viteza exprimată în km/h este egală cu 30 KM/H
Viteza exprimată în mph este egală cu? 18.642 MPH
```

```
import math

r =float(input("Care este viteza in km/h?: "))

print('Viteza exprimata in km/h este egala cu : %0.2f KM/H' %(r))

f = 0.6214 * r

print('Viteza exprimata in mph este egala cu? %0.3f MPH' %(f))
```

Raspuns:

```
Care este viteza in km/h?: 30
Viteza exprimata in km/h este egala cu : 30.00 KM/H
Viteza exprimata in mph este egala cu? 18.642 MPH
```

Ex.2_2

Studiu individual. În acest exercițiu la răspuns trebuie să obținem calendarul pentru această lună. Drept exemplu vom lua luna septembrie din anul 2021. Pentru a obține acest rezultat aveți nevoie de următorul modul `calendar`, care trebuie importat în Python (Sugestie: `calendar.month(anul, luna)`). Cum trebuie să arate codul?, pentru ca la rezultat să obținem:

```
import calendar

print(calendar.month(2022, 3))
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Raspuns:

```
March 2022
Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

Ex2_3

Exercițiul curent este continuarea celui precedent, trebuie să modificăm codul astfel încât anul și luna care ne interesează să fie introduse de către utilizator. Să presupunem că dorim să aflăm care a fost ziua săptămânii în care ne-am născut. Eu am ales în acest exemplu anul și luna mea de naștere și din calendarul apărut pot să aflu această informație.

```
import calendar

r =int(input("Care este anul care va intereseaza?: "))
m =int(input("Care este luna care va intereseaza?: "))
print(calendar.month(r, m))
```

Raspuns:

```
Care este anul care va intereseaza?: 1980
Care este luna care va intereseaza?: 04
April 1980
Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
```

Ex2_4

În acest exercițiu simplu se cere ca utilizatorul să introducă variabilele a și b , iar la răspuns să obținem rezultatul de la operațiile de adunare, scădere, înmulțire și împărțire.

```
import math

a =float(input("Care este valoarea pentru nr.a? a = "))
b =float(input("Care este valoarea pentru nr.b? b = "))

print('a + b = %i' %(a + b))
```

```
print('a - b = %i' %(a - b))
print('a * b = %i' %(a * b))
print('a : b = %0.16f' %(a / b))
```

Raspuns:

```
Care este valoarea pentru nr.a? a = 4
Care este valoarea pentru nr.b? b = 6
a + b = 10
a - b = -2
a * b = 24
a : b = 0.6666666666666666
```

Ex2_6

În acest exercițiu trebuie să cerem de la utilizator care este temperatura în grade Celsius, iar la răspuns să obținem cât este temperatura în grade Celsius și cât este această temperatură în grade Fahrenheit. Dacă ați uitat relațiile de transformare revedeți Ex1_11 din Capitolul 1.

```
import math
r =float(input("Care este temperatura in grade Celsius: "))
f = (r * 9/5) + 32
print('Temperatura in grade Celsius este %0.f grade C' %(r))
print('Temperatura in grade Fahrenheit este %0.1f grade F' %(f))
```

Raspuns:

```
Care este temperatura in grade Celsius: 27
Temperatura in grade Celsius este 27 grade C
Temperatura in grade Fahrenheit este 80.6 grade F
```

Ex2_7

În acest exercițiu trebuie să scriem un cod care să ceară de la utilizator lungimea și lățimea unui dreptunghi, și programul să calculeze aria și perimetrul dreptunghiului cu aceste valori.

```

import math

print(' *****PROGRAM*****')

print('      ARIA & PERIMETRUL      ')

print('      DREPTUNGHI      ')

a =float(input("Care este lungimea dreptunghiului?: "))
b =float(input("Care este latimea dreptunghiului?: "))

p = 2*(a*b)

ar = a * b

print("\nAria = {:.1f} (cm)".format(ar))

print("\nPerimetrul = {:.1f} (cm)".format(p))

```

Raspuns:

```

*****PROGRAM*****
      ARIA & PERIMETRUL
      DREPTUNGHI
Care este lungimea dreptunghiului?: 6
Care este latimea dreptunghiului?: 4

Aria = 24.0 (cm)

Perimetrul = 48.0 (cm)

```

Ex2_8

În acest exercițiu va trebui să scriem un cod care să ceară de la utilizator raza unui cerc și programul să calculeze diametrul, lungimea circumferinței și aria cercului: $D = 2R$; $L = 2\pi R$; $A = \pi R^2$. Ca să nu ne complicăm putem aproxima $\pi \cong 3.14$. Dar în Python avem la dispoziție modulul math de care trebuie să vă folosiți atunci când veți scrie codul pentru acest exercițiu.

```

import math

print(' |||||PROGRAM|||')

print('      CERCUL      ')

print('      LUNGIME & ARiA      ')

a =float(input("Care este raza cercului?: "))

d = 2*a

l = 3.14 * 2 * a

```

```

ar = 3.14 * pow(a, 2)
print("\nAria cercului = {:.1f} (cm)".format(ar))
print("\nDiamentrul cercului = {:.1f} (cm)".format(d))
print("\nLungimea cercului = {:.1f} (cm)".format(l))

```

Raspuns:

```

|||||||PROGRAM|||||||
      CERCUL
    LUNGIME & ARiA
Care este raza cercului?: 5

Aria cercului = 78.5 (cm)

Diamentrul cercului = 10.0 (cm)

Lungimea cercului = 31.4 (cm)

```

Ex2_9

Analizați cu atenție codul și încercați să depistați care sunt erorile comise. La rularea acestui cod vor apărea mesaje de eroare. Sarcina dumneavoastră este să înlăturați greșelile depistate astfel încât codul să ruleze fără erori.

```

nume = "Andrei"
Varsta = 37
Angajat = True
print(nume, Varsta, Angajat)

```

Raspuns:

```

Andrei 37 True

```

Ex2_10

În acest exercițiu se cere să verificați corectitudinea codului și dacă depistați erori să le eliminați astfel încât codul să ruleze fără erori.

```

x = 20
y = 4
print(x - y)

```


Raspuns:

16

Ex2_11

În acest exercițiu trebuie să îl întrebați pe utilizator **Ce planuri ai pentru diseară?** cu ajutorul funcției `input ()`. Răspunsul tastat de utilizator trebuie să îl prezentați sub formă de listă cu ajutorul comenzii `split ()`.

```
a = input('Ce planuri ai pentru diseara?: ')
a1 = a.split()
print("Utilizatorul a introdus cuvintele : ",a1)
```

Raspuns:

```
Ce planuri ai pentru diseara?: diseara plec acasa
Utilizatorul a introdus cuvintele : ['diseara', 'plec', 'acasa']
```

Ex2_12

În acest exercițiu trebuie să îl întrebați pe utilizator **Care este mâncarea ta preferată?** cu ajutorul funcției `input ()`. Răspunsul tastat de utilizator trebuie să îl prezentați doar cu litere majuscule cu ajutorul comenzii `upper ()`.

```
a = input('Care este mancarea ta preferata?: ')
a1 = a.upper()
print("Mancarea ta preferata este : ",a1)
```

Raspuns:

```
Care este mancarea ta preferata?: pizza
Mancarea ta preferata este : PIZZA
```

Ex2_13

În acest exercițiu trebuie să îi cereți utilizatorului să introducă adresa sa de e-mail (în format **prenume.num@utm.md**), iar la răspuns să fie afișat doar numele său (sau ce este scris înaintea simbolului `@`).

```
a = input('Introdu adresa de e-mail: ')
a1 = a.split('@')
print(a1[0])
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Raspuns:

```
Introdu adresa de e-mail: catalin.popa@isa.utm.md
catalin.popa
```

Ex2_14

Vom complica exemplul precedent Ex2_13. La fel vom cere utilizatorului să introducă adresa sa de email (în format **prenume.num@utm.md**) dar la răspuns să obținem mesajul *Salut Prenume Nume*. După cum observați prenumele și numele sunt tastate cu litere mici, noi trebuie să obținem răspunsul cu prima literă majusculă de la prenume și nume.

```
a = input('Introdu adresa de e-mail: ')
a1 = a.split('@')
prenume_num = a1[0].split('.')
nume = prenume_num [0].capitalize()
prenume = prenume_num [1].capitalize()
print('Salut', nume, prenume)
```

Raspuns:

```
Introdu adresa de e-mail: catalin.popa@isa.utm.md
Salut Catalin Popa
```

Ex2_15

În acest exercițiu cerem de la utilizator să introducă prenumele, numele și vârsta cu ajutorul funcției input ().

```
a = input('Introdu prenumele: ')
a0 = input('Introdu numele: ')
a2 = input('Introdu varsta: ')
print('Salut', a, a0, 'ai implinit', a2, 'ani')
```

Raspuns:

```
Introdu prenumele: Catalin
Introdu numele: Popa
Introdu varsta: 20
Salut Catalin Popa ai implinit 20 ani
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Ex2_16

În acest exercițiu trebuie să îi cereți utilizatorului să introducă adresa sa de e-mail (în format **prenume.num@utm.md**), iar la răspuns trebuie să obținem din câte caractere este format numele și prenumele din adresa de e-mail. Sugestie: trebuie să folosiți comanda *len* ().

```
a = input('Introdu adresa de e-mail: ')
a1 = a.split('@')
print("Prenumele si numele din adresa contin () caractere",
      format(len(a1[0])))
```

Raspuns:

```
Introdu adresa de e-mail: catalin.popa@isa.utm.md
Prenumele si numele din adresa contin () caractere 12
```

Ex2_17

În acest exemplu să presupunem că avem definită variabila scriitor = 'Liviu Rebreanu'. La răspuns vrem să apară doar prenumele Liviu. Din cele studiate până în prezent putem obține acest răspuns în două moduri.

```
scriitor = 'Liviu Rebreanu'
prenume = scriitor.split()
#Metoda 1
print(prenume[0])
#Metoda 2
print(scriitor[0:6])
```

Raspuns:

```
Liviu
Liviu
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Laborator nr.3

Ex3_1

Dacă avem lista *nume* = ['Ion', 'Olga', 'Vicu', 'Ana', 'Eva', 'Vlad', 'Stas'].

```
nume = ['Ion', 'Olga', 'Victor', 'Ana', 'Eva', 'Vlad', 'Stas']
for i in nume:
    if i == 'Ion' or i == 'Olga' or i == 'Stas':
        print(i)
```

Raspuns:

```
Ion
Olga
Stas
```

Ex3_2

Dacă avem lista *nume* = ['Ion', 'Olga', 'Vicu', 'Ana', 'Eva', 'Vlad', 'Stas']. Scrieți codul care după rulare să ne dea următorul răspuns:

```
['Ion', 'Olga', 'Stas']
```

```
nume = ['Ion', 'Olga', 'Victor', 'Ana', 'Eva', 'Vlad', 'Stas']
del nume[2]
del nume[2]
del nume[2]
del nume[2]
print(nume)
```

Raspuns:

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

```
['Ion', 'Olga', 'Stas']
```

Ex3_3

Dacă avem lista *nume* = ['Ion', 'Olga', 'Vicu', 'Ana', 'Eva', 'Vlad', 'Stas']. Scrieți codul care după rulare să ne dea următorul răspuns:

a)

```
['Ion', 'Olga', 'Vicu', 'Ana', 'Eva', 'Vlad', 'Stas']
```

b)

```
['Olga', 'Vicu', 'Ana']
```

c)

```
['Ion', 'Olga', 'Vicu']
```

d)

```
['Eva', 'Vlad', 'Stas']
```

e)

```
['Ion', 'Vicu', 'Eva', 'Stas']
```

```
nume = ['Ion', 'Olga', 'Victor', 'Ana', 'Eva', 'Vlad', 'Stas']  
print(nume[0:8])
```

Raspuns:

```
['Ion', 'Olga', 'Victor', 'Ana', 'Eva', 'Vlad', 'Stas']
```

```
nume = ['Ion', 'Olga', 'Victor', 'Ana', 'Eva', 'Vlad', 'Stas']  
print(nume[1:4])
```

Raspuns:

```
['Olga', 'Victor', 'Ana']
```

```
nume = ['Ion', 'Olga', 'Victor', 'Ana', 'Eva', 'Vlad', 'Stas']  
print(nume[0:3])
```

Raspuns:

```
['Ion', 'Olga', 'Victor']
```

```
nume = ['Ion', 'Olga', 'Victor', 'Ana', 'Eva', 'Vlad', 'Stas']  
print(nume[4:])
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Raspuns:

```
['Eva', 'Vlad', 'Stas']  
  
nume = ['Ion', 'Olga', 'Victor', 'Ana', 'Eva', 'Vlad', 'Stas']  
print(nume[::2])
```

Raspuns:

```
['Ion', 'Victor', 'Eva', 'Stas']
```

Ex3_4

Dacă avem lista *lista_mea* = ['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']. Scrieți codul care după rulare să ne dea următorul răspuns:

a)

```
Petru
```

b)

```
['Ion', 'Vlad', 'Petru']
```

c)

```
Ala
```

```
nume = ['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']  
print(nume[1][2])
```

Raspuns:

```
Petru
```

```
nume = ['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']  
print(nume[1])
```

Raspuns:

```
['Ion', 'Vlad', 'Petru']
```

```
nume = ['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']  
print(nume[-1])
```

Raspuns:

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Ala

Ex3_5

Dacă avem lista `lista_mea = ['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']`. Scrieți codul care după rulare să ne dea următorul răspuns:

a) Să obținem lungimea listei (numărul de elemente din listă) cu ajutorul, cu ajutorul funcției `len()`;

b) Să adăugăm elementul `'Iulian'` la sfârșitul listei. Răspunsul va arăta în felul următor:

```
['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala', 'Iulian']
```

c) Să adăugăm elementul `'Iulian'` la începutul listei. Răspunsul va arăta în felul următor:

```
['Iulian', 'Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']
```

```
nume = ['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']  
print (len(nume[0:7]))
```

Raspuns:

4

```
nume = ['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']  
nume.append('Iulian')  
print (nume)
```

Raspuns:

```
['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala', 'Iulian']
```

```
nume = ['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']  
nume2 = ['Iulian']  
nume2.extend(nume)  
print (nume2)
```

Raspuns:

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

```
['Iulian', 'Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']
```

Ex3_6

Dacă avem lista `lista_mea = ['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']`. Scrieți codul în 3 moduri diferite care după rulare să elimine elementul 'Ana' din listă. După rulare trebuie să obținem următorul răspuns:

```
['Vera', ['Ion', 'Vlad', 'Petru'], 'Ala']
```

```
nume = ['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']
nume.remove('Ana')
print (nume)
```

Raspuns:

```
['Vera', ['Ion', 'Vlad', 'Petru'], 'Ala']
```

Ex3_7

Dacă avem lista `my_list = [77, 11, 22, 99, 33, 55]`. Scrieți un cod care la răspuns să ne dea această listă aranjată în ordine crescătoare, descrescătoare și inversată. Folosiți `.sort()`, `.reverse()` și `.sort(reverse = True)`

```
my_list = [77, 11, 22, 99, 33, 55]
my_list.sort()
print (my_list)
```

Raspuns:

```
[11, 22, 33, 55, 77, 99]
```

Ex3_8

Dacă avem lista `old_list = [[22, 44, 66], [11, 33, 55], [20, 30, 40]]`. Scrieți un cod care să creeze o nouă listă `new_list` care să cuprindă doar ultimul element din fiecare sub-listă, în cazul nostru `[66, 55, 40]`

```
old_list = [[22, 44, 66], [11, 33, 55], [20, 30, 40]]
new_list = []
new_list.append(old_list[0][-1])
```



```
new_list.append(old_list[1][-1])
new_list.append(old_list[2][-1])
print(new_list)
```

Raspuns:

```
[66, 55, 40]
```

Ex3_9

Dacă avem *lista_mea = [1,2,3,4,5,6,7,8,9,10]*. Scrieți un cod care să aranjeze într-un mod aleatoriu elementele din listă. Studiu individual, ca sugestie trebuie să folosim (importăm) modulul *random* și să folosim una din comenzile *.randrange()* *.randint()* sau *.shuffle()*

```
my_list = [1,2,3,4,5,6,7,8,9,10]
random.shuffle(my_list)
print(my_list)
```

Raspuns:

```
[4, 8, 5, 6, 9, 2, 3, 10, 7, 1]
```

Ex3_10

Trebuie să creați o listă care să conțină numele celor mai buni 5 prieteni, după care să aranjăm aceste nume în ordine alfabetică.

```
my_list = ['Catalina', 'Ionel', 'Cristinel', 'Pupsic', 'Bogdan']
print(sorted(my_list))
```

Raspuns:

```
['Bogdan', 'Catalina', 'Cristinel', 'Ionel', 'Pupsic']
```

Ex3_11

Trebuie să creați un dicționar care să cuprindă numele a trei prieteni (la cheie) și vârsta acestora (la valoare). La răspuns trebuie să obținem doar vârsta prietenului celui de al doilea prieten din dicționar.

```
my_list = {'Catalina': 18, 'Ionel': 20, 'Cristinel': 40}
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

```
print(my_list['Ionel'])
```

Raspuns:

20

Ex3_12

Folosind dicționarul de la Ex3_11 trebuie să obținem la răspuns media vârstei celor trei prieteni

```
my_list = {'Catalina': 18, 'Ionel' :20, 'Cristinel': 40}
media = (my_list['Catalina'] + my_list['Ionel'] + my_list['Cristinel'])/3
print(media)
```

Raspuns:

26.0

Ex3_13

Folosind dicționarul de la Ex_12 trebuie să mai adăugați 2 prieteni (la cheie) și vârsta lor (la valoare) după care să determinați media vârstei celor 5 prieteni.

```
my_list = {'Catalina': 18, 'Ionel' :20, 'Cristinel': 40}
my_list['Eu'] = 20
my_list['Tu'] = 41
media = sum(my_list.values())/5
print(media)
```

Raspuns:

27.8

Ex3_14

Să presupunem că avem următorul dicționar. Continuați codul astfel încât la răspuns să obținem produsul celor trei valori ale cheilor din dicționar.

```
culori ={'Verde': 3,
        'Alb': 4,
        'Maro': 5}
```

```
my_list = {'Verde': 3, 'Alb' :4, 'Maro': 5}
produs = (my_list['Verde'] * my_list['Alb'] * my_list['Maro'])
print(produs)
```

Raspuns:

				GC-Popa Catalin		Coal
M0	Coal		Semnat	Data		

Ex3_15

Să presupunem că avem următoarea listă:

```
student_note = [
    {'student_nr': 1, 'fizica': 5, 'mate': 7, 'chimia': 6},
    {'student_nr': 2, 'fizica': 8, 'mate': 10, 'chimia': 6},
    {'student_nr': 3, 'fizica': 6, 'mate': 7, 'chimia': 8}
]
```

După cum se observă lista noastră are 3 elemente care sunt dicționare. Trebuie să continuăm codul astfel ca la răspuns să obținem o nouă listă în care elementele sunt dicționare care ne arată media la cele trei discipline pentru fiecare student.

```
student_note = [
    {'student_nr1' :1, 'fizica': 5, 'mate' :7, 'chimia' :6},
    {'student_nr2' :2, 'fizica': 8, 'mate' :10, 'chimia' :6},
    {'student_nr2' :3, 'fizica': 6, 'mate' :7, 'chimia' :8}]

media_1 =
(student_note[0]['fizica']+student_note[0]['mate']+student_note[0]['chimia'])
/3

media_2 =
(student_note[1]['fizica']+student_note[1]['mate']+student_note[1]['chimia'])
/3

media_3 =
(student_note[2]['fizica']+student_note[2]['mate']+student_note[2]['chimia'])
/3

media_student = [
    {'student_nr1' :1, 'media' : media_1},
    {'student_nr2' :2, 'media' : media_2},
    {'student_nr2' :3, 'media' : media_3},]

print(media_student)
```

Raspuns:

```
[{'student_nr1': 1, 'media': 6.0}, {'student_nr2': 2, 'media': 8.0}, {'student_nr2': 3, 'media': 7.0}]
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Ex3_16

Să presupunem că avem următoarea listă:

```
student_note = [  
    {'student_nr': 1, 'fizica': 5, 'mate': 7},  
    {'student_nr': 2, 'fizica': 8, 'mate': 10},  
    {'student_nr': 3, 'fizica': 6, 'mate': 7},  
]
```

Trebuie să scriem un cod astfel încât în fiecare element din listă (fiecare dicționar) să mai apară un element (cheie: valoare) care să conțină media și valoarea numerică a acestei medii.

```
student_note = [  
    {'student_nr1' :1, 'fizica': 5, 'mate' :7},  
    {'student_nr2' :2, 'fizica': 8, 'mate' :10},  
    {'student_nr2' :3, 'fizica': 6, 'mate' :7}]  
  
student_note[0]['media'] =  
(student_note[0]['fizica']+student_note[0]['mate'])/2  
  
student_note[1]['media'] =  
(student_note[1]['fizica']+student_note[1]['mate'])/2  
  
student_note[2]['media'] =  
(student_note[2]['fizica']+student_note[2]['mate'])/2  
  
from pprint import pprint  
pprint(student_note)
```

Raspuns:

```
[{'fizica': 5, 'mate': 7, 'media': 6.0, 'student_nr1': 1},  
 {'fizica': 8, 'mate': 10, 'media': 9.0, 'student_nr2': 2},  
 {'fizica': 6, 'mate': 7, 'media': 6.5, 'student_nr2': 3}]
```

				GC-Popa Catalin		Coal
M0	Coal	Semnat	Data			

Ex3_17

Să presupunem că avem următorul dicționar:

```
my_dict = {'C1': [2,4,6],  
           'C2': [3,5,7]}
```

Trebuie să modificăm astfel acest dicționar astfel încât toate cifrele de la valorile cheilor să fie ridicate la pătrat.

```
my_dict = {'C1': [2,4,6],  
           'C2': [3,5,7]}
```

```
my_dict['C1'] = [my_dict['C1'][0]**2, my_dict['C1'][1]**2,  
my_dict['C1'][2]**2]  
my_dict['C2'] = [my_dict['C2'][0]**2, my_dict['C2'][1]**2,  
my_dict['C2'][2]**2]  
print(my_dict)
```

Raspuns:

```
{'C1': [4, 16, 36], 'C2': [9, 25, 49]}
```

Ex3_18

Să presupunem că avem următorul dicționar:

```
salariu_ora = {'Alex': 27, 'Petru': 29, 'Ion': 30}
```

Trebuie să scriem codul astfel încât după rulare să obținem cheile și valorile din dicționar sub formă de listă și de asemenea să obținem și suma valorilor. Răspunsul trebuie să arate în felul următor:

```
salariu_ora = {'Alex': 27, 'Petru': 29, 'Ion': 30}  
print(salariu_ora)  
salariu_ora= (salariu_ora['Alex']+salariu_ora['Petru']+salariu_ora['Ion'])  
from pprint import pprint  
pprint(salariu_ora)
```

Raspuns:

```
{'Alex': 27, 'Petru': 29, 'Ion': 30}  
86
```

Ex3_19

Să presupunem că avem următorul dicționar:

Scrieți codul care la răspuns să ne dea numărul elementelor din dicționar și să aranjeze elementele cheilor în ordine crescătoare.

Răspunsul trebuie să arate în felul următor:

```
marfa = {'flori': 50, 'fructe': 40, 'legume': 42}
lungimea = len(marfa)
sortare = sorted(marfa.values())
print(lungimea)
print()
print(sortare)
```

Raspuns:

```
3
[40, 42, 50]
```

Ex3_20

Pentru dicționarul din exercițiul precedent scrieți un cod care la răspuns să ne dea doar elementele din dicționar care au valoarea mai mare de 40. Acest exercițiu se poate ușor rezolva cu ajutorul buclei *for* și instrucțiunii de comparare *if* pe care noi le vom studia în capitolul următor. Acest exercițiu îl puteți trece cu vederea dacă nu reușiți să îl rezolvați.

```
marfa = {'flori': 50, 'fructe': 40, 'legume': 42}
marf = ((key, val) for key, val in marfa.items() if val > 40)
print(marf)
```

Raspuns:

```
{'flori': 50, 'fructe': 40, 'legume': 42}
```

Ex3_21

În acest exercițiu trebuie să vă dați seama ce răspuns vom obține după rularea codului următor?

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

```

x = 22
y = 14

print (x < y)
print (x > y)
print (x != y)
print (x == y)

```

```

x = 22
y = 14

print (x < y)
print (x > y)
print (x != y)
print(x == y)

```

Raspuns:

```

False
True
True
False

```

Ex3_22

Corecetați greșelile din cod astfel ca după rulare să obținem *True*.

```

d1 = {'a': 100, 'b': 200, 'c': 300}
d2 = {'a': 300, 'b': 200, 'c': 100}

print(d1['b'] == d2['b'])

# Răspunsul va fi:
# True

```

Raspuns:

Adaugam „'” la d2

Ex3_23

Pentru tupla *t = ('Iulian', 41)* greșelile scrieți un cod care după rulare să afișeze următorul răspuns: *Salut Iulian, ai împlinit 41 de ani!!!*

```

t = ('Iulian', 41)

print("Salut ", t[0], ", au implinit ", t[1], " de ani!!!")

```

Raspuns:

```
Salut Iulian , au implinit 41 de ani!!!
```

Ex3_24

Creați o tuplă care să conțină numele a cinci prieteni. În continuare scrieți codul care după rulare să

ne dea următoarele răspunsuri:

- a) Să fie afișate toate elementele;**
- b) Să fie afișate doar primele 2 elemente din tuplă;**
- c) Să fie afișate doar primele 3 elemente din tuplă.**

```
names = ("Pupsic", "Ion", "Grigore", "Cornel", "Liuba")
for x in range(0, len(names)):
    print(names[x])
print("-----")
for x in range(0,1):
    print(names[x])
print("-----")
for x in range(0,2):
    print(names[x])
```

Raspuns:

```
Pupsic
Ion
Grigore
Cornel
Liuba
-----
Pupsic
-----
Pupsic
Ion
```

Ex3_25

Având un tip de date care conține numele, prenumele și vârsta, exemplu $t = ('Malcoci', 'Iulian', 41)$ determinați nr. de elemente și tipul.

					GC-Popa Catalin		Coal
M0	Coal		Semnat	Data			


```
t = ('Malcoci', 'Iulian', 41)
print(len(t), type(t))
```

Raspuns:

```
3 <class 'tuple'>
```

Ex3_26

Având următoarea tuplă $t = ('Mai', [1,9,20], 'Iunie', (7,11,23))$ scrieți codul care după rulare să ne dea următorul răspuns:

```
20
Mai
11
[1, 9, 20]
(7, 11, 23)
```

```
t = ('mai', [1, 9, 20], 'iunie', (7, 11, 23))
print(t[1][2])
print(t[0])
print(t[3][1])
print(t[1])
print(t[3])
```

Raspuns:

```
20
mai
11
[1, 9, 20]
(7, 11, 23)
```

Ex3_27

În acest exercițiu avem 2 coduri dintre care unul după rulare va afișa o eroare, iar al doilea va rula. Trebuie să examinați ambele coduri și să înțelegeți de ce apare eroarea.

```
a)
t = (27, 7, 9, [11, 20])
t[2] = 10
print(t)
b)
t = (27, 7, 9, [11, 20])
t[3][0] = 10
print(t)
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Raspuns:

TUPLES can't be changed

Ex3_28

Ca și în exemplul precedent avem 2 coduri care după rulare vor da eroare. Trebuie să examinați ambele variante și să înțelegeți de ce apar aceste erori.

```
a)
t = (27, 7, 9, [11, 20])
del t[1]
print(t)

b)
t = (27, 7, 9, [11, 20])
del t
print(t)
```

Raspuns:

Tuples are unchangeable, meaning that we cannot change, add or remove items after the tuple has been created.

Ex3_29

Având următoarea tuplă *discipline* = ('fizica', 'chimia', 'geografia') trebuie să verificăm dacă elementele 'fizica' și 'Fizica' se află în tupla *discipline*.

```
t = ('fizica', 'chimia', 'geografia')
f_mare = 'Fizica'
f_mic = 'fizica'
for x in range(0, len(t)):
    if f_mic == t[x]:
        print(f_mic, " ---- se intalneste")
    else:
        if f_mare == t[x]:
            print(f_mare, " ----- se intalneste")
```

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		

Raspuns:

Fizica ---- se intalneste

Ex3_30

Având următoarea listă *lista_1* = [3, '4', 7, 4, 3, '4', 11] scrieți un cod care după rulare să creeze o listă *lista_2* din care să fie omise elementele duplicate

```
lista_1 = [3, '4', 7, 4, 3, '4', 11]
lista_2 = list(set(lista_1))
print(lista_2)
```

Raspuns:

```
[3, 4, 7, 11, '4']
```

Ex3_31

Având următorul set *setul_meu* = {'ion', 'ana', 'ion', 'vera'} scrieți un cod care să adauge elementul 'vlad' în set, după care să transformați setul în listă, iar după rulare să apară lista.

```
names = {'ion', 'ana', 'ion', 'vera'}
names.add('vlad')
names_list = names.copy()
print(names_list)
```

Raspuns:

```
{'vlad', 'ion', 'vera', 'ana'}
```

Ex3_32

În acest exercițiu sunt prezentate instrucțiunile condiționate if și elif. Trebuie să copiați codul să îl rulați și să încercați să înțelegeți de ce am obținut unul din acele două răspunsuri.

					GC-Popa Catalin		Coal
M0	Coal		Semnat	Data			

```

set_1 = {'mere', 'pere'}
if('banane' in set_1):
    print('DA, bananele sunt în lista de cumpărături!!!')
elif('banane' not in set_1):
    print('NU, bananele nu sunt în lista de cumpărături!!!')

```

Raspuns:

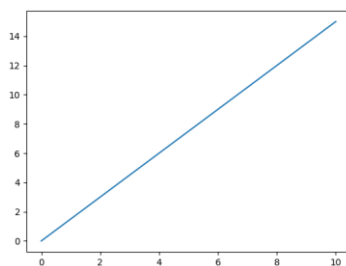
Deoarece in „if” sa verificat daca sunt banana in lista, deoarece nu s-au gasit, s-a executat elif, care este dreapta.

Matplotlib

```

xpoints = np.array([0, 10])
ypoints = np.array([0, 15])
plt.plot(xpoints, ypoints)
plt.show()

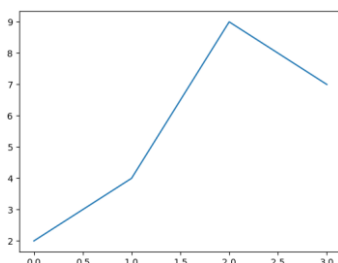
```



```

xpoints = np.array([2, 4, 9, 7])
plt.plot(xpoints)
plt.show()

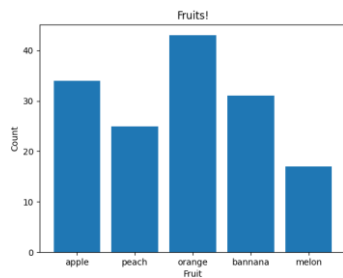
```



```

fruits = ["apple", "peach", "orange", "bannana", "melon"]
counts = [34, 25, 43, 31, 17]
plt.bar(fruits, counts)
plt.title("Fruits!")
plt.xlabel("Fruit")
plt.ylabel("Count")
plt.show()

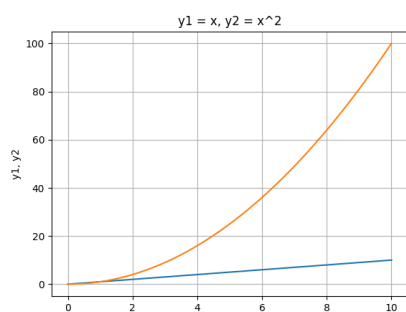
```



```

x = np.linspace(0, 10, 50)
y1 = x
y2 = [i**2 for i in x]
plt.title("y1 = x, y2 = x^2")
plt.xlabel("x")
plt.ylabel("y1, y2")
plt.grid()
plt.plot(x, y1, x, y2)
plt.show()

```



```

import numpy as np
import matplotlib.pyplot as plt

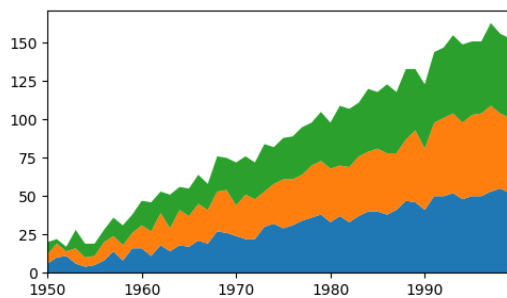
rng = np.arange(50)
rnd = np.random.randint (0, 10, size=(3, rng.size))

```

```

yrs = 1950 + rng
fig, ax = plt.subplots(figsize=(5, 3))
ax.stackplot(yrs, rng + rnd, labels=['Moldova', 'Italia', 'Romania'])
ax.set_xlim(xmin=yrs[0], xmax=yrs[-1])
fig.tight_layout()
plt.show()

```



Python 5.10

Parenthesized context managers

Using enclosing parentheses for continuation across multiple lines in context managers is now supported. This allows formatting a long collection of context managers in multiple lines in a similar way as it was previously possible with import statements. For instance, all these examples are now valid:

New features

macOS arm64 Binaries

ParaView is now tested on the macOS arm64 platform.

Due to this change, ParaView binaries are now available using the processor name according to the platform in the binary filename rather than a generic “32Bit” or “64Bit” indicator.

Linux: [x86_64](#) macOS: [arm64](#) and [x86_64](#) Windows: [AMD64](#)

- Windows users: If installing Python 3.5.1 as a non-privileged user, you may need to escalate to administrator privileges to install an update to your C runtime libraries.
- Windows users: There are now "web-based" installers for Windows platforms; the installer will download the needed software components at installation time.

- Windows users: There are redistributable zip files containing the Windows builds, making it easy to redistribute Python as part of another software package. Please see the documentation regarding [Embedded Distribution](#) for more information.
- Windows users: Some virus scanners (most notably "Microsoft Security Essentials") are flagging "Lib/distutils/command/wininst-14.0.exe" as malware. This is a "false positive": the file does not contain any malware. We build it ourselves, from source, on a known-clean system. We've asked that this false positive report be removed, and expect action soon. In the meantime, please don't be alarmed to see this warning when installing Python 3.5.2, or when scanning any earlier version of 3.5.
- OS X users: The OS X installers are now distributed as signed installer package files compatible with the OS X Gatekeeper security feature.

Better error messages

SyntaxErrors

When parsing code that contains unclosed parentheses or brackets the interpreter now includes the location of the unclosed bracket of parentheses instead of displaying *SyntaxError: unexpected EOF while parsing* or pointing to some incorrect location. For instance, consider the following code (notice the unclosed '{'):

NameErrors

When printing [NameError](#) raised by the interpreter, `PyErr_Display()` will offer suggestions of similar variable names in the function that the exception was raised from:

Declarative approach

Readers may be aware of pattern matching through the simple example of matching a subject (data object) to a literal (pattern) with the switch statement found in C, Java or JavaScript (and many other languages). Often the switch statement is used for comparison of an object/expression with case statements containing literals.

More powerful examples of pattern matching can be found in languages such as Scala and Elixir. With structural pattern matching, the approach is “declarative” and explicitly states the conditions (the patterns) for data to match.

While an “imperative” series of instructions using nested “if” statements could be used to accomplish something similar to structural pattern matching, it is less clear than the “declarative” approach. Instead the “declarative” approach states the conditions to meet for a match and is more readable through its explicit patterns. While structural pattern matching can be used in its simplest form comparing a variable to a literal in a case statement, its true value for Python lies in its handling of the subject’s type and shape.

Simple pattern: match to a literal

Let's look at this example as pattern matching in its simplest form: a value, the subject, being matched to several literals, the patterns. In the example below, `status` is the subject of the match statement. The patterns are each of the case statements, where literals represent request status codes. The associated action to the case is executed after a match:

Patterns and classes

If you are using classes to structure your data, you can use as a pattern the class name followed by an argument list resembling a constructor. This pattern has the ability to capture class attributes into variables:

					GC-Popa Catalin	Coal
M0	Coal		Semnat	Data		