

## Structuri de control în limbajul Python.

### Instrucțiunile FOR și WHILE

#### Obiectivele lucrării de laborator



1. Familiarizarea cu sintaxa structurilor de control în Python
2. Execuția și analiza scripturilor prezentate
3. Aplicabilitatea corectă a noțiunilor prezentate prin rezolvarea unor probleme asemănătoare cu cele exemplificate în cadrul laboratorului

#### Instrucțiunea FOR

for variabila in lista:  
    bloc de instructiuni  
    [break]  
    [continue]  
else:  
    bloc de instructiuni

#### Ex. 1 – Afișarea primelor n numere naturale

```
Thonny - D:\Didactic\2022-2023\PCLP2\new labs\Lab 01\for.py @ 2:17
File Edit View Run Tools Help
[Icons]
for.py x
1 n=int(input('Introduceti o valoare pentru n:'))
2 for i in range(n):
3     print(i, end=' ')

Shell x
>>> %Run for.py
Introduceti o valoare pentru n:15
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
>>>
```

#### Exerciții propuse:

1. Executați și analizați scriptul de la Ex. 1.
2. Modificați ex. 1 astfel încât să se afișeze:
  - a. Primele n numere naturale în ordine descrescătoare
  - b. Toate numerele impare în intervalul (0,n]
  - c. produsul primelor n numere naturale (factorialul unui număr).
  - d. Suma numerelor din intervalul (0,n] divizibile cu un număr x citit de la tastatură
3. Se citește de la tastatura un număr natural. Să se calculeze suma cifrelor sale.
4. Scrieți un program pentru a afișa următorul model folosind bucla **for**.

```

*
**
***
****
*****
*****
*****
****
***
**
*

```

5. Scrieți un program pentru a calcula suma seriilor unui număr  $x$ , citit de la tastatură, până la  $n$  termeni. De exemplu, dacă  $x=3$  și  $n=5$  seria va deveni  $3 + 33 + 333 + 3333 + 33333 = 37035$

### Instrucțiunea WHILE

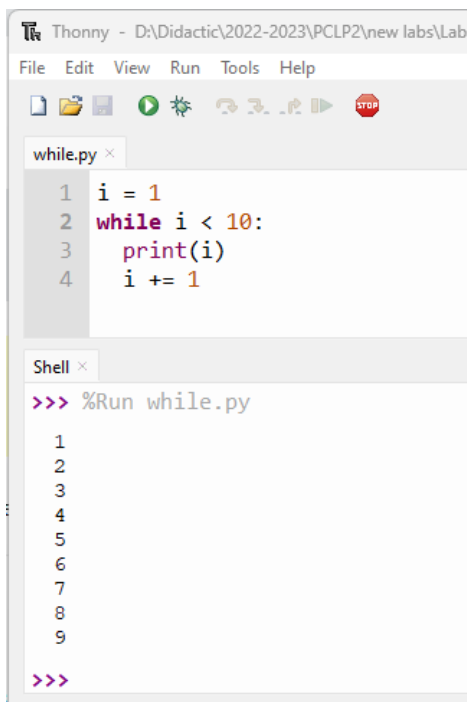
---

```

while conditie:
    bloc de instructiuni
    [break]
    [continue]
else:
    bloc de instructiuni

```

#### Ex. 2 – Afișarea primelor 10 numere naturale



The screenshot shows the Thonny Python IDE interface. The main editor window displays a Python script named 'while.py' with the following code:

```

1 i = 1
2 while i < 10:
3     print(i)
4     i += 1

```

Below the editor is a 'Shell' window showing the execution output. The command '%Run while.py' has been executed, resulting in the numbers 1 through 9 being printed on separate lines. The prompt '>>>' is visible at the bottom of the shell window.

## Ex. 3 – Validarea introducerii unui număr într-un interval dat

```
1. nr = int(input('Introdu un numar intre 100 si 500: '))
2. while nr < 100 or nr > 500:
3.     print('Numarul este incorect! Mai incearca...')
4.     nr = int(input('Introdu un numar intre 100 si 500: '))
5. else:
6.     print("Numarul introdus este corect!")
```

## Exerciții propuse:

1. Să se citească de la tastatură un șir de numere încheiat cu numărul 0 și să se calculeze suma lor.
2. Scrieți un program pentru a introduce un număr și afișați reversul acestuia  
ex. Pentru X=123 se va afișa numărul 321.
3. Să se scrie un program care calculează și afișează valoarea polinomului  $p(x)=3x^2-7x-10$  pentru  $x=1,2,...,10$ . Valorile polinomului vor fi afișate ca mai jos:  
 $p(1)=3*1^2-7*1-10=-14$   
 $p(2)=3*2^2-7*2-10=-12$   
 $p(3)=3*3^2-7*3-10=-4$   
....

**Intrucțiunea selectiva**

Începând cu versiunea Python 3.10, echivalentul instrucțiunii switch din limbajul C este instrucțiunea **match-case**. Analizați modul de lucru cu această instrucțiune în documentația Python (<https://peps.python.org/pep-0636/>), apoi implementați următoarea aplicație de tip meniu cu următoarele opțiuni.

Operații cu numere întregi:

[C] citire 2 nr întregi

[+] adunare

[-] scădere

[\*] înmulțire

[/] împărțire

[X] exit

## Laborator 3

