

## TUTORIAT 1

Linkuri utile:

- <https://www.python.org/>
- <https://docs.python.org/3/>
- <https://www.codecademy.com/learn/learn-python-3>
- <https://www.learnpython.org/>
- <https://stanfordpython.com/#/>

Ce conține tutoriatul ?

- I. Despre limbajul python
- II. Noțiuni introductive
  1. Variabile, tipuri de date
  2. Operatori
  3. Instrucțiuni
  4. Citire și afișare de la tastatură
- III. Exerciții

### ***I.Despre limbajul python***

Numele limbajului provine de la serialul BBC “Monthly Python’s Flying Circus” (Cercul zburător al șarpelui Monty); serial pe care Guido van Rossum (inițiatorul implementării limbajului python) îl urmărea și nu are nici o legătură cu reptilele.

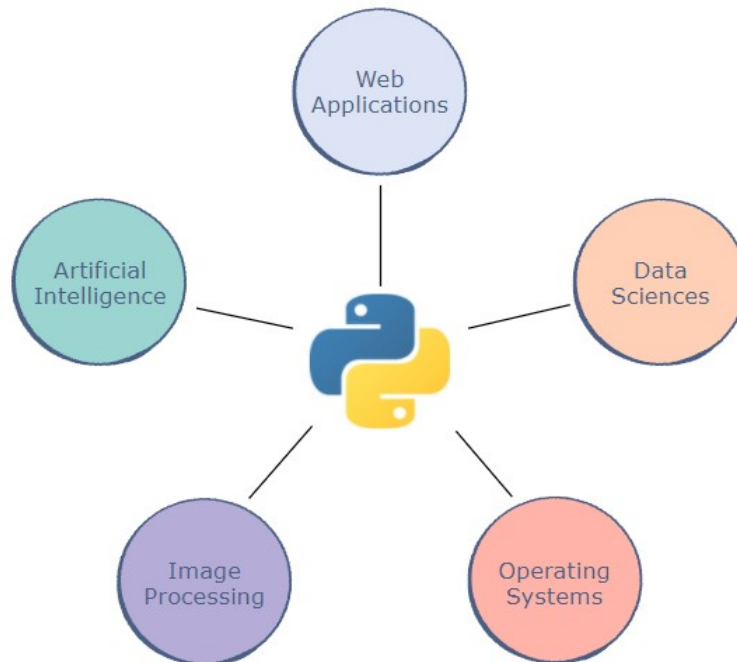
Prima versiune a fost lansată în anul 1994 (python 1.0), urmând și alte versiuni. În anul 2008 a fost lansată versiunea 3.0, iar ultima versiune Python este 3.10 ( lansată pe 4 octombrie 2021).

Limbajul python propune o sintaxă simplă și este unul dintre cele mai populare limbaje de programare din lumea modernă. Este un limbaj high-level ( nu poate fi înțeles direct de către o mașină). Ca orice limbaj de programare, python aduce cu sine atât avantaje, cât și dezavantaje. Printre avantaje se numără: limbaj portabil, open-source, cu o sintaxă sugestivă, potrivită pentru începătorii în programare. Unele dintre dezavantaje sunt reprezentate de: lipsa verificării tipurilor de date la compilare, este mai lent în comparație cu alte limbaje.

Este un limbaj interpretat (comparat cu C/C++ compilat). Interpretarea se oprește la prima eroare, în urma interpretării este distribuită o sursă, limbajul este portabil, interpretatorul e dependent de platformă.



Pentru ce este utilizat ?



## II. Noțiuni introductive

Despre sintaxă la modul general (comparație cu C++)

C++	Python 3.x
blocuri delimitate prin {}	blocuri delimitate prin indentare
instrucțiuni finalizate cu ;	este necesar ; doar la scrierea pe același rând a mai multor instrucțiuni
comentarii: //, /*linii de comentat*/	comentarii: #, ``` linii de comentat ``` , " " "linii de comentat " " "

### II.1. Variabile, tipuri de date

#### Declarare variabilă în C++

```
int    a    =    5;
tipul  numele  valoare
```

#### Declarare variabilă în Python

```
a    =    5
nume  valoare
```

#### Concluzie:

Variabilele în python au tip dinamic (preiau tipul obiectului pe care îl reprezintă, variabilele sunt de fapt referințe spre obiecte). Tipul variabilei este stabilit la inițializare. Atât

valoarea, cât și tipul unei variabile se pot schimba de-a lungul programului. Tipul unei variabile poate fi verificat cu ajutorul funcției `type(ume_variabila)`.

**Tipuri de date existente:** `int`, `float`, `complex (a+bj)`, `bool`, `NoneType`; `list`, `tuple`, `str` (mutabile); `set`, `frozenset` (imutabile), dicționare. (**!ATENȚIE! NU EXISTĂ TIPUL CHAR**)

## II.2.Operatori

Caracteristică importantă: aritatea -> numărul de operanzi căruia îi este aplicat

Precedența operatorilor: <https://docs.python.org/3/reference/expressions.html#operator-precedence>

Operația numerică	Sintaxa operației	Sintaxa atribuirii
Adunare	$x + 3$	$x += 3$
Scădere	$x - 3$	$x -= 3$
Înmulțire	$x * 3$	$x *= 3$
Împărțire	$x / 3$	$x /= 3$
Împărțire întreagă	$x // 3$	$x //= 3$
Restul împărțirii	$x \% 3$	$x \% = 3$
Ridicare la putere	$x ** 3$	$x ** = 3$

Operația booleană	Sintaxa operației
Not	Not x
And	X and y
Or	X or y
Egal (diferit)	$A == b$ ( $a != b$ )
Mai mare (sau egal)	$A > b$ ( $a >= b$ )
Mai mic (sau egal)	$A < b$ ( $a <= b$ )
Expresie înlănțuită	$a > b > c$

## II.3. Instrucțiuni

Instrucțiunea condițională **if** (nu există **switch**)

**if** expresie :

instrucțiune1

**else:**

instrucțiune2

**if** expresie :

instrucțiune1

**elif** expresie:

instrucțiune2

**else:**

instrucțiune3

Instrucțiunea repetitivă cu test inițial **while** (nu există do while)

**while** expresie:

instructiune

Instrucțiunea **for**

**for** nume\_variabilă **in** colecție\_iterabilă:

my\_list = [1, 8, "Ana"]

**for** element **in** my\_list:

# element = 1, 8, "Ana"

**for** i **in** range(10):

# i = 0, 1, 2, ..., 9

**range([min], max, [pas])**

- ➔ Parametrii min și pas nu sunt obligatorii (default min = 0, pas = 1)
- ➔ Se generează pe rând valorile dintre **min** și **max** ( fără valoarea max, adică numerele din intervalul [min, max)
- ➔ **range(x)** => [0,1,..., x - 1]
- ➔ **range(x,y)** => [x, x+1, x+2,...,y-1]
- ➔ **range(x,y,pas)** => [x, x+pas, x+2\*pas, ...]

**Clauza else** (pentru instrucțiunile repetitive for, while)

**for** i **in** range(x):

# instructiuni

**else:**

#instructiuni

Clauza **else** se executa dacă instrucțiunea s-a terminat fără o ieșire forțată (cu **break**).

## II.4.Citirea și afișarea de la tastatură

funcția **print()**

- ➔ are mai mulți parametri separați prin virgulă
- ➔ valorile transmise afișate implicit pe același rând, separate prin spațiu, urmate de o linie nouă
- ➔ putem modifica afișarea implicită prin parametri **sep** și **end**

functia **input ()**

- ➔ parametru opțional un mesaj
- ➔ returnează șirul de caractere introdus până la sfârșitul de linie ( rezultat de tip string => necesită conversie dacă dorim alt tip)

### **III.Exerciții**

- 1.Să se determine dacă 3 numere reale a, b, c (citite de la tastatură) pot fi lungimile laturilor unui triunghi.
2. Se citește un număr de la tastatură. Să se afișeze dacă este sau nu cub perfect.
3. Se citește un număr întreg n. Să se efectueze următoarele acțiuni condiționate:

- dacă n este **impar**, afișați **Weird**
- dacă n este **par** și este **inclus** în intervalul **[2,5]**, afișați **Not Weird**
- dacă n este **par** și este **inclus** în intervalul **[6,20]**, afișați **Weird**
- dacă n este **par** și **mai mare** decât **20**, afișați **Not Weird**

Exemplu: n = 24 => Not Weird ( n > 20 , n par)

4. Să se determine primul număr prim din intervalul [a,b], cu a și b citite de la tastatură. (folosind clauza else).
- 5.Se citesc două numere naturale a și b ( a < b). Să se afișeze numerele din intervalul [a,b] cu proprietatea că au număr maxim de divizori primi. Exemplu: a = 30, b = 45 => 30, 42 (au câte trei divizori primi, iar restul numerelor au mai puțini).
6. Se citește un șir format din n numere întregi ( n > 2). Să se afișeze cele mai mici două valori distincte din șir sau mesajul „Nu se poate”, dacă acestea nu există.
- 7.Să se calculeze și să se afișeze valoarea funcției definite prin:

$$f(x) = \begin{cases} |x| & , x \leq -9 \\ \sqrt{x+9} & , -9 < x < 0 \\ x^2 & , 0 \leq x < 10 \end{cases}$$