

Fundamentele programării

Elemente de bază ale unui program Python

```
a = input("Number: ")
b = input("Number: ")
c = int(a) + int(b)
print(f"The sum of {a} + {b} is {c}")
```

Comentarii

- încep cu `#` și țin până la sfârșitul liniei
- încep cu `"""` și țin mai multe rânduri, până la `"""`

Obiecte

Toate datele într-un program Python - obiecte

Un obiect are:

- o identitate - adresa lui în memorie
- un tip - care determină operațiile posibile și valorile pe care le poate lua
- o valoare

Odată creat, identitatea și tipul obiectului nu mai pot fi modificate. Valoarea unor obiecte se poate modifica:

- obiecte mutabile - se poate modifica
- obiecte imutabile - nu se pot modifica, orice operație creează un nou obiect

Tipuri de date

int:

- numere întregi (pozitive și negative)
- operații: $+$, $-$, $*$, $/$, $//$, $**$, $\%$, $==$, $!=$, $<$, $>$, operații pe biți: $|$, $^$, $\&$, $<<$, $>>$, \sim
- literali: 1 , -3

bool:

- valori True și False
- operații: and , or , not
- literali: False, True, 0 , 1

float:

- numere reale
- operații: $+$, $-$, $*$, $/$, $==$, $!=$, $<$, $>$
- literali: 3.14

NoneType

- o singură valoare: None
- operații: $==$, $!=$
- literal: None

Secvențe:

- dacă a este o secvență atunci:
 - $\text{len}(a)$ returnează numărul de elemente;
 - $a[0], a[1], \dots, a[\text{len}(a)-1]$ sunt elementele lui a.
- operații: $=$, $+$, $*$

String: 'abc', "abc" este o secvență imutabilă

List: [2, 3], [1, 'a', [1, 3]] - secvență mutabilă

Tuple: (2, 3), (1, 'a', (1, 3)) - secvență imutabilă

Dictionar: {'num': 1, 'demon': 2}

- conține perechi (cheie - valoare)
- operația de găsim a unei valori după cheie este eficientă
- orice cheie apare o singură dată

Funcție:

- funcțiile în Python pot fi tratate ca orice alt tip de valoare

List

- creare [7,9]
- accesare valori , modificare valori , verificare dacă un element este în listă (2 în [1, 2, 'a'])
- ștergere inserare valori (append, insert, pop)
- slicing, liste izotrope
- listele se pot folosi în for

```
# create
a = [1, 2, 'a']
print (a)
x, y, z = a
print(x, y, z)

# indices: 0, 1, ..., len(a) - 1
print (a[0])
print ('last element = ', a[len(a)-1])
# lists are mutable
a[1] = 3
print (a)
```

```
# lists as stacks
stack = [1, 2, 3]
stack.append(4)
print (stack)
print (stack.pop())
print (stack)
```

```
#generate lists using range
l1 = range(10)
print (list(l1))
l2 = range(0,10)
print (list(l2))
l3 = range(0,10,2)
print (list(l3))
l4 = list(range(9,0,-1))
print (l4)
```

```
# slicing
print (a[:2])
b = a[:]
print (b)
b[1] = 5
print (b)
a[3:] = [7, 9]
print(a)
a[:0] = [-1]
print(a)
a[0:2] = [-10, 10]
print(a)
```

```
# nesting
a = [1, [1, 1, 9], 9]
print (a)
b = [1, 1, 9]
c = [1, b, 9]
print (c)
```

```
#list in a for loop
l = range(0,10)
for i in l:
    print (i)
```

Tuple

- creare (23, 34, 2)
- sunt secvente imutabile
- se pot folosi in for
- unpacking

<pre># Tuples are immutable sequences # A tuple consists of a number # of values separated by commas # tuple packing t = 12, 21, 'ab' print(t[0]) # empty tuple (0 items) empty = ()</pre>	<pre># tuple with one item singleton = (12,) print (singleton) print (len(singleton)) # tuple in a for t = 1, 2, 3 for el in t: print (el)</pre>
<pre># sequence unpacking x, y, z = t print (x, y, z)</pre>	<pre># Tuples may be nested u = t, (23, 32) print(u)</pre>

Dictionar

- creare {} sau {'num': 1, 'denom': 2}
- sunt secvente imutabile
- stergere per cheie (cheie, valoare)

<pre>#create a dictionary a = {'num': 1, 'denom': 2} print(a) #get a value for a key print(a['num'])</pre>	<pre>#set a value for a key a['num'] = 3 print(a) print(a['num'])</pre>
<pre>#delete a key value pair del a['num'] print (a)</pre>	<pre>#check for a key if 'denom' in a: print('denom = ', a['denom']) if 'num' in a: print('num = ', a['num'])</pre>

Instructiuni

• Atribuire / Legare

- $x = 1$
- $x = x + 2$
- modificare secvență:

$x = 1$

- relegare de nume:

$x = x + 2$

- modificare secvență:

$y = [1, 2]$

$y[0] = -1$

• Blocuri

- se realizează prin indentarea liniilor

Instructiuni - If, While

```
if conditie:
    bloc de instructiuni
elif conditie:
    bloc de instructiuni
else:
    bloc de instructiuni
```

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

```
def gcd(a, b):
    """
    Return the greatest common divisor of two positive integers.
    """
    if a == 0: return b
    if b == 0: return a

    while a != b:
        if a > b:
            a = a - b
        else:
            b = b - a
    return a

print (gcd(7,15))
```

Instructions - For

```
for el in secventa: #parcurem element cu element
    bloc de instrucțiuni #el - element in secvența
    [break]
    [continue]
else:
    bloc de instrucțiuni #executat daca s-a dat break
```

```
#use a list literal
for i in [2,-6,"a",5]:
    print (i)

#using a variable
x = [1,2,4,5]
for i in x:
    print (i)

#using range
for i in range(10):
    print (i)

for i in range(2,100,7):
    print (i)

#using a string
s = "abcde"
for c in s:
    print (c)
```


Parcurgere în Python

Pythonic	Programator C++/Java/C#/Pascal
<pre>for i in range(6): print (i)</pre>	<pre>for i in [0,1,2,4,5]: print (i)</pre>
<pre>x = [2,-6,"a",5] for el in x: print (el)</pre>	<pre>x = [2,-6,"a",5] for i in range(len(x)): print (x[i])</pre>
<pre>x = [2,-6,"a",5] for el in reversed(x): print (el)</pre>	<pre>x = [2,-6,"a",5] for i in range(len(x)-1,-1): print (x[i])</pre>
<pre>x = [2,-6,"a",5] for i, el in enumerate (x): print (i, "->", el)</pre>	<pre>x = [2,-6,"a",5] for i in range(len(x)): print (i, "->", x[i])</pre>
<pre>#parcure 2 liste simultan x = [2,-6,"a",5] y = [2,-6,"a"] for elx, ely in zip (x,y): print (elx, "<->", ely)</pre>	<pre>#parcure 2 liste simultan x = [2,-6,"a",5] y = [2,-6,"a"] n = min(len(x),len(y)) for i in range(n): print (x[i],y[i])</pre>

Dictionar: dictionar = { 'num': 1, 'demon': 2 }

<pre>#parcure cheile din dictionar for cheie in dictionar: print (cheie)</pre>
<pre>#parcure cheile din dictionar for cheie in dictionar.keys(): print (cheie)</pre>
<pre>#parcure valorile din dictionar for valoare in dictionar.values(): print (valoare)</pre>
<pre>#parcure perechile din dictionar for cheie, valoare in dictionar.items(): print (cheie, valoare)</pre>