

ΕΙΣΑΓΩΓΗ
στον
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ
με
PYTHON

ΓΙΩΡΓΟΣ ΦΑΚΙΟΛΑΚΗΣ
2024

ΠΕΡΙΛΗΨΗ

ΒΑΣΙΚΟΙ ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ

Τύπος	Ονομασία
int	integer (ακέραιος αριθμός)
float	floating point (αριθμός κινητής υποδιαστολής - δεκαδικός)
str	string (αλφαριθμητικό)
bool	boolean (Λογικός)

ΒΑΣΙΚΟΙ ΤΕΛΕΣΤΕΣ

Αριθμητικοί τελεστές

Τελεστής	Λειτουργία
+	Πρόσθεση αριθμών ή συνένωση αλφαριθμητικών
-	Αφαίρεση αριθμών
*	Γινόμενο αριθμών ή επανάληψη αλφαριθμητικού
/	Διαίρεση
**	Ύψωση σε δύναμη
//	Πηλίκο ακέραιας διαίρεσης
%	Υπόλοιπο ακέραιας διαίρεσης

Τελεστές σύγκρισης και βασικοί Λογικοί τελεστές

Τελεστής	Έκφραση	Εξήγηση
==	$x == y$	x είναι ίσος με y
!=	$x != y$	x είναι διάφορος του y
>	$x > y$	x είναι μεγαλύτερος από y

<	$x < y$	x είναι μικρότερος από y
>=	$x \geq y$	x είναι μεγαλύτερος ή ίσος με y
<=	$x \leq y$	x είναι μικρότερος ή ίσος με y
not	not p	άρνηση πρότασης p
or	p or q	λογική διάζευξη
and	p and q	λογική σύζευξη

ΒΑΣΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΤΥΠΩΝ

α/α	ΣΥΝΑΡΤΗΣΗ	ΛΕΙΤΟΥΡΓΙΑ
1	int()	μετατρέπει συμβολοσειρα ή float (κόβει δεκαδικά) σε ακέραιο
2	float()	μετατρέπει συμβολοσειρα ή int σε πραγματικό
3	str()	μετατρέπει αριθμό σε συμβολοσειρα
4	round()	στρογγυλοποιεί float σε ακέραιο

ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ

Σελ.	ΕΝΤΟΛΗ	ΛΕΙΤΟΥΡΓΙΑ
4	print	εμφανίζει στην οθόνη μηνύματα ή τιμές μεταβλητών
5	input()	διαβάζει από το πληκτρολόγιο τιμές μεταβλητών str ¹
6	=	υπολογίζει τιμές παραστάσεων ή βάζει τιμες σε μεταβλητές
8	if	ελέγχει μια συνθήκη και εκτελεί ένα σύνολο εντολών αν η συνθήκη είναι True και ένα άλλο (ή τίποτα) αν η συνθήκη είναι False
11	while	εκτελεί πολλές φορές ένα σύνολο εντολών, όσο ισχύει μια συνθήκη
13	for	εκτελεί συγκεκριμένο αριθμό επαναλήψεων ενός συνόλου εντολών, για τις τιμές της μεταβλητής σε καθορισμένο σύνολο τιμών

¹ αν θέλουμε να εισάγουμε άλλου τύπου τιμές πχ ακέραιες ή πραγματικές, πρέπει να τις μετατρέψουμε χρησιμοποιώντας αντίστοιχες συναρτήσεις όπως int(), float() κλπ.

ΕΝΤΟΛΕΣ

Έξοδος: print

Εμφανίζει στην οθόνη μηνύματα ή τιμές μεταβλητών

Σύνταξη

`print('μήνυμα', μεταβλητή,...)`

Λειτουργία

Εμφανίζει στην οθόνη μηνύματα, μέσα σε εισαγωγικά ή και τιμές μεταβλητών.

Παράδειγμα

```
>>> print('Hello World')
Hello World
>>> print (name)
Γιώργο
>>> print('με λένε',name)
με λένε Γιώργο
>>>
```

Παρατήρηση 1: Μπορεί να χρησιμοποιηθούν τελεστές για υπολογισμό παραστάσεων ή και συνενώσεις ή επαναλήψεις αλφαριθμητικών.

```
>>> print(2+3)
5
>>> print('είμαι ο '+'Κώστας')
είμαι ο Κώστας
>>>
```

*Παρατήρηση 2: Μπορεί να χρησιμοποιηθούν χαρακτήρες διαφυγής. Υπάρχουν χαρακτήρες οι οποίοι δεν εμφανίζονται αλλά προκαλούν κάποια ενέργεια, πχ ο χαρακτήρας νέας γραμμής (enter). Για να χειριστεί τέτοιους χαρακτήρες σε συμβολοσειρές, η Python διαθέτει τους χαρακτήρες διαφυγής (escape characters) που αρχίζουν με το χαρακτήρα *

\\	backslash
\'	μονά εισαγωγικά
\"	διπλά εισαγωγικά
\n	νέα γραμμή
\t	tab

Παραδείγματα:

```
>>> print('\\Hello\\')
\\Hello\\
>>> print('It\'s OK')
It's OK
>>> |
```

Είσοδος δεδομένων: input()

Διαβάζει από το πληκτρολόγιο αλφαριθμητικές τιμές και τις αποθηκεύει στη μνήμη σε μεταβλητές

Σύνταξη

μεταβλητή=input('μήνυμα')

Λειτουργία

1. σταματάει η εκτέλεση του προγράμματος και περιμένει από τον χρήστη να πληκτρολογήσει κάτι.
2. Όταν ο χρήστης πατήσει το πλήκτρο Enter, η συνάρτηση input βάζει στη μνήμη στην αντίστοιχη μεταβλητή την τιμή που έχει πληκτρολογηθεί σαν συμβολοσειρά και συνεχίζεται η ροή εκτέλεσης του προγράμματος.

Παράδειγμα

```
>>> name=input('πως σε λένε;')
πως σε λένε;Γιώργο
>>> print(name)
Γιώργο
>>> |
```

Παρατήρηση 1: Χρήσιμο είναι να εμφανίζεται και ένα μήνυμα (prompt) που να πληροφορεί τον χρήστη για το είδος της εισόδου που αναμένεται από το πρόγραμμα.

Παρατήρηση 2: Η input() μπορεί να συνδυαστεί με συναρτήσεις μετατροπής της συμβολοσειράς σε αριθμό:

```
>>> x=int(input('δώσε ακέραιο αριθμό '))
δώσε ακέραιο αριθμό 4
>>> print(x+2)
6
>>> |
```

Εντολή απόδοσης τιμής: =

Δημιουργεί μεταβλητές και βάζει τιμές σε αυτές.

Αποτελείται από τρία μέρη:

- το αριστερό που υποχρεωτικά πρέπει να είναι μία μεταβλητή,
- το ίσον = (τελεστής εκχώρησης²) και
- το δεξί μέρος που μπορεί να είναι μία τιμή, μία άλλη μεταβλητή ή μία έκφραση που υπολογίζεται.

Σύνταξη

1. μεταβλητή=τιμή
2. μεταβλητή=παράσταση

Λειτουργία

1. δημιουργεί τη μεταβλητή και της βάζει την τιμή
2. Υπολογίζει την παράσταση και στη συνέχεια δημιουργεί τη μεταβλητή και βάζει το αποτέλεσμα του προηγούμενου υπολογισμού σε αυτήν.

Παράδειγμα

```
>>> x=10
>>> print(x)
10
>>> name='Γιώργος'
>>> print(name)
Γιώργος
>>> A=2*x+3
>>> print(A)
23
>>> A=A+2
>>> print(A)
25
>>> |
```

Παρατήρηση 1: Η Python δίνει την δυνατότητα ταυτόχρονης ανάθεσης τιμών σε μεταβλητές.

```
>>> X=Y=Z=1
>>> print(X,Y,Z)
1 1 1
>>> x,y,name=1,2,'Μήτσος'
>>> print(x,y,name)
1 2 Μήτσος
>>> |
```

² Το ίσον (=) είναι τελεστής εκχώρησης και όχι ισότητας.

ΔΙΑΚΛΑΔΩΣΗ

Συνθήκες: Προτάσεις που μπορούν να γίνουν ΑΛΗΘΕΙΣ (True) ή ΨΕΥΔΕΙΣ (False)

Παραδείγματα

Πρόταση	Τιμή
$2=2$	True
$1+1>=5$	False
$A1 \leq 5$	Αν το A1 περιέχει έναν αριθμό μικρότερο ή ίσο του 5 τότε True. Αν το A1 περιέχει έναν αριθμό μεγαλύτερο του 5 τότε False
$B2+B3>10$	Αν το άθροισμα των τιμών των μεταβλητών B2 και B3 είναι μεγαλύτερο του 10 τότε True, αλλιώς False
$X \geq 10 \text{ and } X < 15$	Αν η τιμή της μεταβλητής X είναι ανάμεσα στο 10 (συμπεριλαμβανομένου) και 15 τότε True, αλλιώς False

Τελεστές σύγκρισης:

(Πχ $A1=2$, $B1=3$)

ΣΥΜΒΟΛΟ	ΠΡΑΞΗ	ΠΑΡΑΔΕΙΓΜΑ	ΑΠΟΤΕΛΕΣΜΑ
$==$	Ίσο με	$A1==B1$	False
$<$	Μικρότερο από	$A1<B1$	True
$\leq$	Μικρότερο ή ίσο	$A1 \leq B1$	True
$>$	Μεγαλύτερο από	$A1>B1$	False
$\geq$	Μεγαλύτερο ή ίσο	$A1 \geq B1$	False
$!=$	διαφορετικό	$A1 != B1$	True

Λογικοί τελεστές

(πχ. Λογικές προτάσεις p , q)

Πίνακας Αληθείας

p	q	$\text{not } p$	$p \text{ and } q$	$p \text{ or } q$
T	T	F	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	F

Εντολή IF

Ελέγχει μια συνθήκη και αν είναι True τότε εκτελεί ένα σύνολο εντολών, αν η συνθήκη είναι False τότε εκτελεί ένα άλλο ή τίποτα.

- Τα σύνολα των εντολών (μπλοκ) καθορίζονται από τις εσοχές (στοίχιση) στη γραφή. Οι εντολές ενός μπλοκ πρέπει να είναι μετατοπισμένες προς τα δεξιά.
- Η τυπική μετατόπιση ή εσοχή (indentation) των εντολών είναι τέσσερα κενά. Η Python μάς βοηθάει σε αυτό ρυθμίζοντας αυτόματα τις εσοχές για μας, απλά πατώντας Enter μετά την πληκτρολόγηση της άνω κάτω τελείας.
- Η έλλειψη ενός κενού ή η ύπαρξη επιπλέον κενών μπορεί να οδηγήσει σε λάθος ή απρόσμενη συμπεριφορά σε ένα πρόγραμμα.
- Η πρώτη εντολή που ευθυγραμμίζεται με το αριστερό περιθώριο είναι η πρώτη εντολή έξω από το μπλοκ.

Σύνταξη

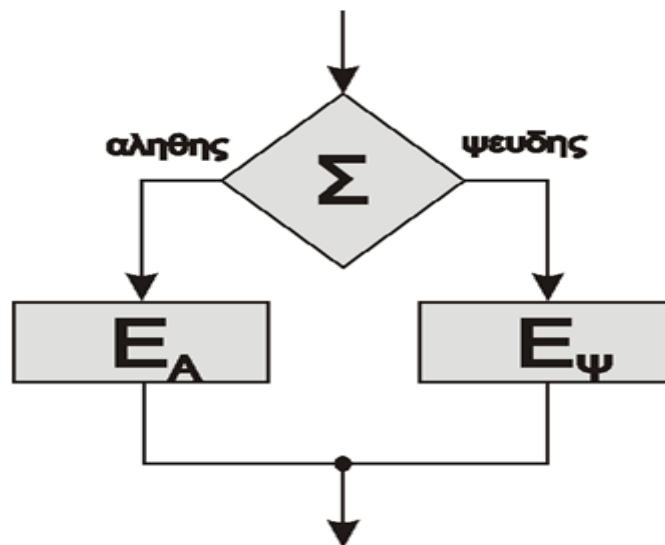
IF συνθήκη :

```
...  
    εντολές  $E_A$ 
```

else :

```
...  
    εντολές  $E_\Psi$   
...
```

Λειτουργία:



Παρατήρηση 1: Στην εντολή IF εκτελείται **μόνο** ο ένας από τους δύο κλάδους.

Παράδειγμα

```
A=10
B=15
MO=(A+B)/2
if MO>=10:
    print('OK')
else:
    print('προσοχή!!!!')
```

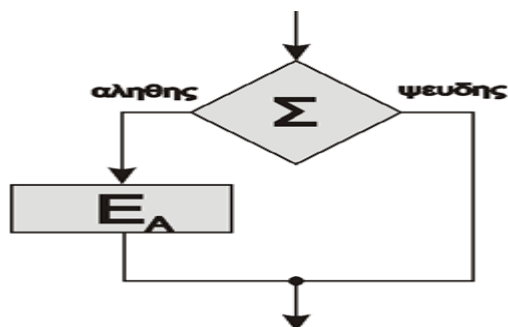
→run→

OK
>>>

...πιο απλή...

IF συνθήκη :

```
...
εντολές ΕΑ
...
```



...ΠΙΟ ΣΥΝΘΕΤΗ...

If συνθήκη1 :

...
εντολές E_1

elif συνθήκη2 :

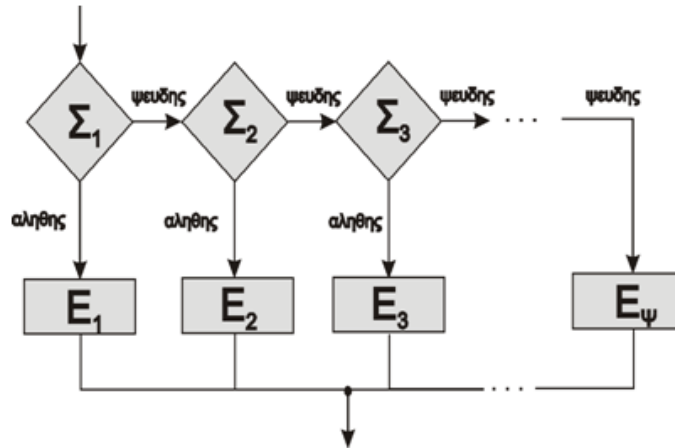
...
εντολές E_2

elif συνθήκη3 :

...
εντολές E_3
...

else:

...
εντολές E_ψ
...



Εμφωλευμένες if

Μπορούμε να έχουμε και εμφωλευμένες if (μία if μέσα σε άλλη if).

Για παράδειγμα:

```
x=10
y=20
if x==y:
    print(x, '=', y)
else:
    if x<y:
        print(x, '<', y)
    else:
        print(x, '>', y)
```

→run→

```
10 < 20
>>>
```

ΕΠΑΝΑΛΗΨΗ

Οι εντολές επανάληψης χρησιμοποιούνται για την εκτέλεση ενός συνόλου εντολών (Ε) πολλές φορές.

Ο αριθμός των επαναλήψεων εξαρτάται από μια συνθήκη (Σ) ή από έναν μετρητή

Εντολή while

Εκτελεί ένα σύνολο εντολών πολλές φορές, όσο ισχύει μια συνθήκη

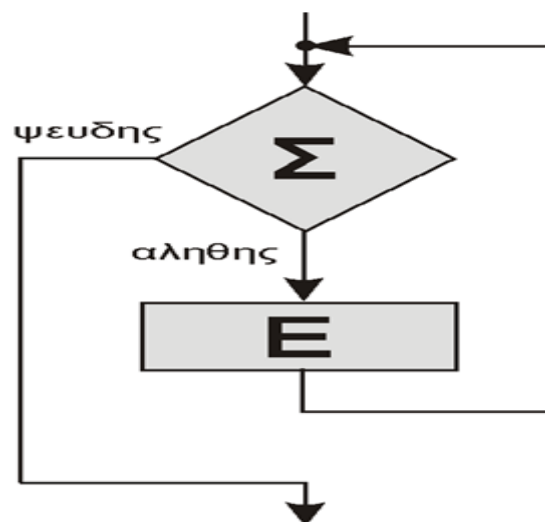
- Τα σύνολα των εντολών (μπλοκ) καθορίζονται από τις εσοχές (στοίχιση) στη γραφή. Οι εντολές ενός μπλοκ πρέπει να είναι μετατοπισμένες προς τα δεξιά.
- Η τυπική μετατόπιση ή εσοχή (indentation) των εντολών είναι τέσσερα κενά. Η Python μάς βοηθάει σε αυτό ρυθμίζοντας αυτόματα τις εσοχές για μας, απλά πατώντας Enter μετά την πληκτρολόγηση της άνω κάτω τελείας.
- Η έλλειψη ενός κενού ή η ύπαρξη επιπλέον κενών μπορεί να οδηγήσει σε λάθος ή απρόσμενη συμπεριφορά σε ένα πρόγραμμα.
- Η πρώτη εντολή που ευθυγραμμίζεται με το αριστερό περιθώριο είναι η πρώτη εντολή έξω από το μπλοκ.

Σύνταξη

while συνθήκη :

```
...  
    εντολές Ε  
...
```

Λειτουργία:



Παρατήρηση 1: Το πλήθος των επαναλήψεων δεν είναι καθορισμένο. Μπορεί να είναι από 0 έως πάρα πολλές.

Παρατήρηση 2: ΑΡΧΙΚΟΠΟΙΗΣΗ: Πριν την εντολή while πρέπει να υπάρχει τουλάχιστον μια εντολή αρχικοποίησης των μεταβλητών της συνθήκης

Παρατήρηση 2: ΤΕΡΜΑΤΙΣΜΟΣ: Μέσα στις εντολές της while πρέπει να υπάρχει τουλάχιστον μια εντολή που να τροποποιεί τις τιμές των μεταβλητών της συνθήκης έτσι ώστε κάποια στιγμή αυτή να γίνει «False» για να τερματίσει η επανάληψη.

Παράδειγμα

```
i=1
while i<=10:
    print(i, 'Γιώργος')
    i=i+1
```

→run→

```
1 Γιώργος
2 Γιώργος
3 Γιώργος
4 Γιώργος
5 Γιώργος
6 Γιώργος
7 Γιώργος
8 Γιώργος
9 Γιώργος
10 Γιώργος
>>> |
```

Οι break και continue

Η εντολή **break** χρησιμοποιείται για άμεση έξοδο από οποιοδήποτε σημείο ενός βρόγχου

Όταν εκτελεστεί η **continue** στο «σώμα» ενός βρόγχου, ο έλεγχος ροής μεταφέρεται στη συνθήκη του βρόγχου..

Παράδειγμα:

```
sum=0
while True:
    x=int(input('δώσε αριθμό (-1 για τέλος)'))
    if x==-1:
        break
    sum=sum+x
print(sum)
```

→run→

```
δώσε αριθμό (-1 για τέλος)5
δώσε αριθμό (-1 για τέλος)1
δώσε αριθμό (-1 για τέλος)9
δώσε αριθμό (-1 για τέλος)-1
15
>>> |
```

Παρατήρηση: Γενικά δεν είναι καλή πρακτική προγραμματισμού η χρήση των break ή continue.

Εντολή for

Εκτελεί συγκεκριμένο αριθμό επαναλήψεων ενός συνόλου εντολών, για τις τιμές της μεταβλητής σε καθορισμένο σύνολο τιμών

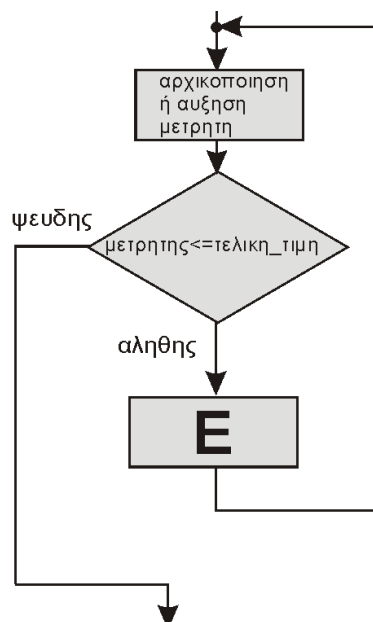
- Τα σύνολα των εντολών (μπλοκ) καθορίζονται από τις εσοχές (στοίχιση) στη γραφή. Οι εντολές ενός μπλοκ πρέπει να είναι μετατοπισμένες προς τα δεξιά.
- Η τυπική μετατόπιση ή εσοχή (indentation) των εντολών είναι τέσσερα κενά. Η Python μάς βοηθάει σε αυτό ρυθμίζοντας αυτόματα τις εσοχές για μας, απλά πατώντας Enter μετά την πληκτρολόγηση της άνω κάτω τελείας.
- Η έλλειψη ενός κενού ή η ύπαρξη επιπλέον κενών μπορεί να οδηγήσει σε λάθος ή απρόσμενη συμπεριφορά σε ένα πρόγραμμα.
- Η πρώτη εντολή που ευθυγραμμίζεται με το αριστερό περιθώριο είναι η πρώτη εντολή έξω από το μπλοκ.

Σύνταξη

for μετρητής in [λίστα τιμών]:

```
...  
    εντολές E  
...
```

Λειτουργία:



Παράδειγμα:

```
for i in [1,3,5,7,9]:  
    print (i)
```

→run→

```
1  
3  
5  
7  
9
```

Συνάρτηση range

Αντί για συγκεκριμένη λίστα τιμών μπορεί να χρησιμοποιηθεί η συνάρτηση **range()** η οποία παράγει λίστες τιμών. Δηλαδή η:

range (αρχική τιμή, τελική τιμή, βήμα)

παράγει τη λίστα τιμών:

[αρχική τιμή, αρχική τιμή + βήμα,... τελική τιμή]

πχ η range(1,10,2) παράγει τη λίστα [1,3,5,7,9]

Δηλαδή μπορούμε να συντάξουμε την εντολή:

for μετρητής in range (αρχική τιμή, τελική τιμή, βήμα):

```
...  
εντολές E  
...
```

Παράδειγμα:

```
for i in range(1,10,3):  
    print (i)
```

→run→

```
1  
4  
7  
>>>
```

Παρατήρηση 1: Η αρχική τιμή και το βήμα μπορεί να παραληφθεί. Σε αυτή την περίπτωση η range() θέτει ως αρχική τιμή το 0 και ως βήμα το 1. Πχ. η range(10) παράγει τη λίστα [0,1,2,3,4,5,6,7,8,9,10]

Παρατήρηση 2: Αν το βήμα έχει αρνητική τιμή τότε η λίστα που παράγεται είναι φθίνουσα.

Παρατήρηση 3: Μπορούν να χρησιμοποιηθούν και η break και η continue, όπως και στην εντολή while.