

ΔΙΑΛΕΞΗ 6^η

Τεχνολογίες και Μεθοδολογίες Προγραμματισμού

Μηχανικών Σχεδίασης Προϊόντων και Συστημάτων Εξάμηνο Β'

- ΕΠΑΝΑΛΗΨΗ ΣΤΙΣ ΜΕΘΟΔΟΥΣ ΤΩΝ ΔΟΜΩΝ ΕΛΕΓΧΟΥ



ΔΟΜΕΣ ΕΠΑΝΑΛΗΨΗΣ Ο ΒΡΟΧΟΣ FOR

Χρησιμοποιείται για επανάληψη μέσω μιας ακολουθίας τιμών.
Η γενική μορφή ενός for βρόχου είναι:

for μεταβλητή **in** ακολουθία:
μπλόκ προτάσεων

Ακολουθία μπορεί να είναι ακολουθία αριθμών, μία συμβολοσειρά, μια λίστα, μια πλειάδα ή ένα αντικείμενο αρχείου.

Στην μεταβλητή εκχωρείται διαδοχικά κάθε τιμή της ακολουθίας.

Το μπλοκ προτάσεων εκτελείται μετά από κάθε εκχώρηση.

Κάθε εσοχή λέει στον διερμηνευτή, που αρχίζει και που τελειώνει το μπλοκ.

ΔΟΜΕΣ ΕΠΑΝΑΛΗΨΗΣ Ο ΒΡΟΧΟΣ WHILE

Η δεύτερη εντολή επανάληψης που διαθέτει η Python, είναι η εντολή **while**. Χρησιμοποιείται όταν δεν είναι γνωστό το πλήθος των δεδομένων.

Η γενική μορφή της while είναι:

While συνθήκη:

----εντολές A #----=τέσσερα κενά ή ένα tab

Όσο η συνθήκη ισχύει (True), εκτελούνται οι εντολές (A) που βρίσκονται με εσοχή μέσα στη while.

Μόλις εκτελεστεί και η τελευταία εντολή, ο έλεγχος περνά και πάλι στην 1^η γραμμή της while, όπου γίνεται ξανά έλεγχος που καθορίζει τη συνέχεια ή το σταμάτημα των επαναλήψεων.

Η ΕΝΤΟΛΗ WHILE

Σημαντικό στοιχείο στη While είναι ότι όλες οι μεταβλητές που συμμετέχουν στη συνθήκη της θα πρέπει:

- Να έχουν αρχική τιμή (συνήθως μηδέν), πριν από αυτήν και έξω από το βρόχο της
- Να ενημερώνονται και να αλλάζουν (τουλάχιστον μία από αυτές), μέσα στο εσωτερικό της, (και συνήθως στο τέλος του βρόχου, αλλά μέσα σ' αυτόν).

ΠΑΡΑΔΕΙΓΜΑ 1

Να γράφει πρόγραμμα που να εμφανίζει τους αριθμούς από το 1 έως και το 5 καθώς και τα τετράγωνά τους

```
num=1  #αρχική τιμή πριν και έξω από το βρόχο  
while num<=5:  
    print (num,num**2)  
    num+=1  #αυξάνει #το num κατά ένα
```

ΠΑΡΑΔΕΙΓΜΑ 1a

Αύξουσα και φθίνουσα εκτύπωση περιττών

```
i=0
```

```
while i <= 4:
```

```
    print(2*i+1) #ο τύπος των περιττών αριθμών
```

```
    i += 1
```

```
print('*****')
```

```
i=9
```

```
while i >= 1:
```

```
    print(i)
```

```
    i += -2 #Αν από το 9 κατεβούμε κατά 2, μας δίνει τους περιττούς αριθμούς
```

ΠΑΡΑΔΕΙΓΜΑ 2 Διακοπή εισαγωγής στοιχείων, όταν ισχύσει συνθήκη

Να γραφεί πρόγραμμα το οποίο αθροίζει αριθμούς μέχρι το άθροισμα να ξεπεράσει το 1000. Να αναφέρει πόσοι αριθμοί χρειάστηκαν, καθώς και το άθροισμά τους

```
s=0 #μεταβλητή αθροίσματος
i=1 #μεταβλητή επαναλήψεων
while s<1000:
    s+=i #Πρόσθεσε στο νέο s την τιμή του i
    i+=1 #Αύξησε το i κατά 1
print('epanalipsis =:',i)
print('athrisma =:',s)
```

ΕΠΕΞΗΓΗΣΗ ΠΑΡΑΔΕΙΓΜΑΤΟΣ 2

Τρέχοντας το πρόγραμμα:

Την **1^η** φορά:

s=0, i=1. Μπαίνει στο βρόχο της While και κάνει τον έλεγχο: Είναι το $0 < 1000$? **Η συνθήκη είναι True**, γι' αυτό και συνεχίζει παρακάτω. Το **s** αυξάνεται κατά **i** δηλαδή $0+1 = 1$ και το **i** αυξάνει κατά **1**, δηλαδή $1+1 = 2$.

Την **2^η** φορά:

s=1, i=2. Μπαίνει στο βρόχο της While και κάνει τον έλεγχο: Είναι το $1 < 1000$? **Η συνθήκη είναι True**, γι' αυτό και συνεχίζει παρακάτω. Το **s** αυξάνεται κατά **i** δηλαδή $1+2 = 3$ και το **i** αυξάνει κατά **1**, δηλαδή $2+1 = 3$.

Την **3^η** φορά:

s=3, i=3. Μπαίνει στο βρόχο της While και κάνει τον έλεγχο: Είναι το $1 < 1000$? **Η συνθήκη είναι True**, γι' αυτό και συνεχίζει παρακάτω. Το **s** αυξάνεται κατά **i** δηλαδή $3+3 = 6$ και το **i** αυξάνει κατά **1**, δηλαδή $3+1 = 4$.

ΠΑΡΑΔΕΙΓΜΑ 3

Να γραφεί πρόγραμμα το οποίο υπολογίζει το μέγιστο, το ελάχιστο, το πλήθος των αριθμών που δόθηκαν καθώς και τη μέση τιμή μιας ακολουθίας μη αρνητικών αριθμών. Για τερματισμό δώστε -1

```
count=0 #μετρητής
sum=0    #άθροισμα
print('δωσε -1 για τερματισμό')
num=eval(input('dose enan arithmo '))
min=num      #αρχικοποίηση
max=num      #αρχικοποίηση
while num!=-1:
    count+=1 #Ξεκίνα και ανέβασε το μετρητή κατά 1
    sum+=num #Αύξησε το άθροισμα με το νούμερο που δώσαμε
    if num<=min:
        min=num
    if num>=max:
        max=num
    num=eval(input('dose enan arithmo '))

print('mikroteros=',min)
print('megaliteros=',max)
print('sum=',sum)
print('plithos arithmon=:',count)
print('mesos oros=',sum/count)#εκτύπωση απλή
print('mesos oros = {0:.2f}'.format(sum/count))
```

Δοκιμάστε την εντολή `num=eval(input('dose enan arithmo '))` ως εξής:

```
num=eval(input(f'dose {count+1}o arithmo '))
```



Η Συνάρτηση range

Η ενσωματωμένη συνάρτηση range χρησιμοποιείται για να παράγει μια αριθμητική ακολουθία αριθμών.

Η γενική της μορφή είναι: `range(αρχή, τέλος, βήμα)`.

Αν **αρχή** = **α**, **τέλος** = **τ**, και **βήμα** ο αριθμός που θα προστεθεί σε κάθε τιμή **=β**, τότε η ακολουθία που παράγεται είναι: $a, a+1, a+2, \dots, \tau-1$.

- ▶ Αν το βήμα β παραλείπεται τότε, $\beta=1$.
- ▶ Αν το a παραλείπεται τότε $a=0$, το τέλος δεν μπορεί να παραλειφτεί.
- ▶ Οι αριθμοί αρχή, τέλος και βήμα πρέπει να είναι ακέραιοι

Ο τελεστής in

Ο τελεστής in χρησιμοποιείται σε πολλές περιπτώσεις **για τον έλεγχο** ύπαρξης μιας τιμής σε μια ακολουθία τιμών, και επιστρέφει πάντα μία τιμή **True ή False**, αλλά αποτελεί και δομικό στοιχείο της εντολής επανάληψης for.

- Ως στοιχείο ελέγχου

```
>>> 5 in [1 2 5 12]
```

```
SyntaxError: invalid syntax
```

```
5 in [1,2,5,12] ή 5 in (1,2,5,12)
```

```
True
```

```
>>> 3 in [1,2,5,12]
```

```
False
```

Παράδειγμα 4

Το παρακάτω πρόγραμμα εμφανίζει 4 ακέραιους και τα τετράγωνά τους

```
for i in range(2,6):  
    print(i,i**2)
```

Παράδειγμα 5

Υποθέστε ότι ο πληθυσμός μιας πόλης είναι 300000 το 2014 και αυξάνεται 3% κάθε χρόνο. Το πρόγραμμα να εμφανίζει τον πληθυσμό μέχρι το 2018

```
pop=300000
print('{0:20}{1}'.format('Year','plithismos'))
for year in range(2014,2019):
    print('{0:<20}{1:,>d}'.format(year, round(pop)))
    pop+=0.03*pop
```

Παράδειγμα 6

Το πρόγραμμα ζητά το ποσό που κατατίθεται σε ένα λογαριασμό και το ετήσιο επιτόκιο. Υπολογίζει το υπόλοιπο του λογαριασμού μετά από τρεις μήνες για τέσσερα τρίμηνα. Δίνεται $telposo = arxposo * (1 + monthlyepit)^i$

```
arxposo=eval(input('Δώσε το αρχικό ποσό που καταθέτεις =?'))
epit=eval(input('Δώσε το ετήσιο επιτόκιο πχ 0.02, 0.03 :=?'))
monthlyepit = (epit/12)
print('{0}{1:>15}'.format('ΜΗΝΑΣ','ΠΟΣΟ'))
for i in range(3,13,3):
    telposo=arxposo*(1+monthlyepit)**i
    print('{0:2}{1:>20,.2f}'.format(i,telposo))
```

Η εντολή **break**

- ▶ Η εντολή **break** προκαλεί έξοδο μέσα από έναν βρόχο. Όταν εκτελείται μέσα σε ένα βρόχο, ο βρόχος τερματίζει αμέσως.
- ▶ Αν δηλαδή στη διάρκεια εκτέλεσης ενός βρόχου αποφασίσουμε ότι θέλουμε να διακόψουμε την εκτέλεσή του, (άρα να μην εκτελεστούν τα επόμενα βήματά του), χρησιμοποιούμε την εντολή **break**
- ▶ Η **break** μπορεί να δοθεί σε οποιοδήποτε σημείο μέσα σε ένα βρόχο **for** ή **while**
- ▶ Οι εντολές **break** συνήθως εμφανίζονται μέσα στις εντολές **if** .

Σύνταξη : `for i in range():`
 `if λογική έκφραση:`
 `break`
 εντολές

Η εντολή **continue**

Όταν εκτελείται η εντολή **continue** μέσα σε έναν βρόχο `for` ή `while`:

- ▶ Παραλείπονται οι υπόλοιπες εντολές μέσα στο βρόχο, και το πρόγραμμα συνεχίζει με την επόμενη επανάληψη.

Σύνταξη: `for i in range ():`

`if λογική έκφραση:`

`continue`

 εντολές

ΠΑΡΑΔΕΙΓΜΑ Continue

```
for i in range(5):  
    if i == 3:  
        continue  
    print(i)  
Print('')  
s=0  
for i in range(1,10):  
    s+=5  
    if s>26:  
        continue  
    print(s,i)
```

ΠΑΡΑΔΕΙΓΜΑ Continue 1

Να γραφεί πρόγραμμα Python που θα ζητάει από τον χρήστη την πληκτρολόγηση ενός κειμένου μέχρι να δοθούν οι χαρακτήρες 'end'. Το πρόγραμμα χρησιμοποιώντας την ενσωματωμένη συνάρτηση len(), (η οποία μετράει το πλήθος των χαρακτήρων ενός αλφαριθμητικού) θα πληροφορεί τον χρήστη εάν το αλφαριθμητικό ήταν πολύ μικρό (<6 χαρακτήρες) ή επαρκές.

```
while True:
```

```
    s=str(input('Εισάγετε κάτι: '))
```

```
    if s == 'end':
```

```
        break
```

```
    if len(s)<6:
```

```
        print ('Πολύ μικρό')
```

```
        continue
```

```
    print ('Το μήκος των εισαχθέντων στοιχείων είναι επαρκές')
```

Παράδειγμα `continue` 2

Να γραφεί πρόγραμμα που τυπώνει από 1 έως 9. Όταν ο αριθμός είναι πολλαπλάσιο του 3, να βγαίνει μήνυμα

```
for i in range(1,10):  
    if i%3==0:  
        print(i,'Πολλαπλάσιο του 3')  
        continue  
    print(i)
```

Παράδειγμα 7α (Χρήση της for με break)

Το πρόγραμμα ψάχνει για ένα συγκεκριμένο στοιχείο σε μία λίστα.

Διακόπτεται μόνο του, όταν βρεθεί το στοιχείο

```
numbers=(1,8,7,4,11,12,2,9,2,5)
```

```
search=11
```

```
for number in numbers:
```

```
    if search==number:
```

```
        print('Το βρήκα',(numbers))
```

```
        break
```

```
else:
```

```
    print('den yparxei')
```

Παράδειγμα 7b (Χρήση της for με break)

Να γράφει πρόγραμμα που δέχεται μέχρι 10 αριθμούς και να σταματά αν το άθροισμά τους είναι μεγαλύτερο από 100. **Διακόπτεται μόνο του, όταν βρεθεί το ζητούμενο**

```
s=0
```

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

```
    x=int(input('Δώσε έναν ακέραιο := '))
```

```
    s+=x
```

```
    if s>100:
```

```
        break
```

Παράδειγμα 7c (Χρήση της while με break)

Να γραφεί πρόγραμμα που δέχεται μέχρι 10 αριθμούς και να σταματά αν το άθροισμά τους είναι μεγαλύτερο από 100. **Διακοπή εισαγωγής στοιχείων, όταν δεν τηρείται κάποια συνθήκη**

```
i=s=0
```

```
while True: #επανέλαβε για πάντα
```

```
    x=int(input('Δώσε έναν ακέραιο :'))
```

```
    s+=x
```

```
    i+=1
```

```
    if s>100 or i==10:
```

```
        break
```

Παράδειγμα 7d Διακοπή εισαγωγής στοιχείων από το χρήστη .

Να γραφεί πρόγραμμα Python που θα διαβάσει πολλούς αριθμούς και θα εμφανίσει τον διπλάσιο του καθενός. Η εισαγωγή αριθμών σταματά όταν ο χρήστης πληκτρολογήσει τον αριθμό 0 (Χρήση της while με break)

```
while True: #επανέλαβε για πάντα
    num = int(input('Δώσε έναν αριθμό:'))
    if num == 0:
        break
    print (num*2)
```

ΠΑΡΑΔΕΙΓΜΑ 8 Διακοπή εισαγωγής στοιχείων από το χρήστη

Να γραφτεί πρόγραμμα που θα δέχεται μέχρι 10 αριθμούς. Θα σταματά αμέσως αν δοθεί η τιμή 0 και θα εμφανίζει αν αυτό συνέβη ή όχι

i=0

while i<10: # Αν δοθούν μη μηδενικοί αριθμοί, η break δεν θα εκτελεστεί, ο βρόχος θα τελειώσει λόγω της συνθήκης while και αμέσως μετά θα εκτελεστεί ο κώδικας που βρίσκεται στο else. Αντίθετα αν δοθεί μηδέν, #εκτελείται η break και το τμήμα else αγνοείται

x=int(input('Δώσε έναν ακέραιο αριθμό : = '))

if x==0:

print('Δόθηκε 0. Άρα σταματώ το πρόγραμμα.')

break

else:

print('Δεν δόθηκε 0. Συνέχισε ')

i+=1

ΠΑΡΑΔΕΙΓΜΑ 8b Διακοπή εισαγωγής στοιχείων από το χρήστη
ΠΑΡΑΔΕΙΓΜΑ 4 b Διακοπή εισαγωγής στοιχείων από το χρήστη θα σταματά όταν δοθεί αρνητικός αριθμός, οπότε θα τυπώνει το πλήθος των αριθμών που δόθηκαν.

```
i=0
pl=0
while i<10:
    x=int(input('Δώσε έναν ακέραιο αριθμό : = '))
    if x==0:
        print('Δόθηκε 0. Άρα σταματώ το πρόγραμμα.')
        break
    i+=1
    pl+=1
print('Δόθηκαν ',pl,'αριθμοί')
```

#Στο τέλος κάθε επανάληψης η input θα δεχτεί μία τιμή, η οποία θα τυπωθεί στην επόμενη επανάληψη ή θα καθορίσει τον τερματισμό των επαναλήψεων.



ΠΑΡΑΔΕΙΓΜΑ 8c Έλεγχος εγκυρότητας 1

Να δοθεί ένας αριθμός από το 0 έως το 9

```
number=int(input('Δώσε έναν αριθμό από το 0 έως το 9: '))  
while number < 0 or number > 9:  
    number=int(input('Σου είπα από το 0 έως το 9: '))  
print('Έδωσες τον αριθμό ' + str(number))
```