



Python – Μάθημα 5

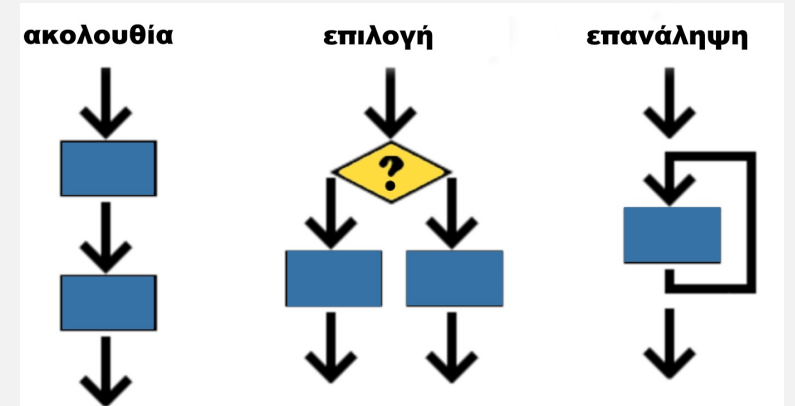
Δομή Επανάληψης – for - while



Δομή Επανάληψης

Όταν θέλουμε ένα τμήμα του προγράμματος να επαναληφθεί πολλές φορές τότε κάνουμε χρήση της **δομής επανάληψης**.

Από τις εντολές που μας προσφέρει η python για τη δομή επανάληψης εμείς θα ασχοληθούμε με τη **for** και τη **while**.



- Η **for** χρησιμοποιείται μόνο όταν έχουμε **γνωστό αριθμό επαναλήψεων**
- Η **while** χρησιμοποιείται όταν έχουμε **είτε γνωστό είτε άγνωστο αριθμό επαναλήψεων**.



For

Σύνταξη:

```
for x in range(start, stop, step):  
    ομάδα εντολών 1
```

Η for εκτελεί μια ομάδα εντολών (**ομάδα εντολών 1**) πολλές φορές. Για να το πετύχει αυτό χρησιμοποιεί μια μεταβλητή (**x**) που συνήθως παίρνει τιμές με τη βοήθεια της συνάρτησης **range()**.

Παράδειγμα:

Κώδικας

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

Εκτέλεση

```
>>> %Run -c $EDITOR_CONTENT  
1  
2  
3  
4  
5  
6  
7  
8  
9
```



range()

Για να καταλάβουμε καλύτερα πως λειτουργεί ο προηγούμενος κώδικας πρέπει να κατανοήσουμε τη χρήση της **range()**

range(start, stop, step)

Η συνάρτηση range() επιστρέφει μια ακολουθία αριθμών με βάση τις 3 παραμέτρους που είναι μέσα στις παρενθέσεις της.

- Η πρώτη παράμετρος (**start**) ορίζει την πρώτη τιμή που επιστρέφεται . (προεπιλογή στο 0).
- Η δεύτερη παράμετρος (**stop**) καθορίζει το τέλος της ακολουθίας (αποκλειστική)
- Η τρίτη παράμετρος (**step**) καθορίζει το βήμα με το οποίο μεταβάλλονται οι τιμές που επιστρέφονται. (προεπιλογή 1).



range()

Παραδείγματα κλήσης της range:

- **range(1,11,1)** επιστρέφει τις τιμές [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- **range(1,10,2)** επιστρέφει τις τιμές [1, 3, 5, 7, 9]
- **range(10,20,1)** επιστρέφει τις τιμές [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
- **range(10,1,-1)** επιστρέφει τις τιμές [10, 9, 8, 7, 6, 5, 4, 3, 2]
- **range(5,10)** ισοδυναμεί με range(5,10,1) επιστρέφει τις τιμές [5, 6, 7, 8, 9]
- **range(7)** ισοδυναμεί με range(0,7,1) επιστρέφει τις τιμές [0, 1, 2, 3, 4, 5, 6]



For

Τώρα που έχουμε εξηγήσει πως λειτουργεί η `range()` είναι πιο εύκολο να κατανοήσουμε και τον κώδικα που είδαμε στη διαφάνεια 3.

Η μεταβλητή `x` παίρνει τις τιμές που επιστρέφει η **`range(1,10,1)`**. Επομένως στην πρώτη επανάληψη η τιμή του `x` είναι 1 και με την **`print`** εμφανίζεται η τιμή αυτή. Εφόσον το βήμα της `range` είναι 1, η τιμή του `x` θα **αυξάνεται κατά 1** κάθε φορά και αυτό θα συνεχίζεται όσο η τιμή του `x` είναι **μικρότερη (ΠΡΟΣΟΧΗ όχι ίση) του 10**.

Συνεπώς οι τιμές που εμφανίζονται κατά την εκτέλεση του προγράμματος είναι οι: 1, 2, 3, 4, 5, 6, 7, 8, 9.

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

```
>>> %Run -c $EDITOR_CONTENT
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9
```




Δραστηριότητα 5.1 – διψήφιοι περιττοί

Να γράψετε πρόγραμμα που να εμφανίζει όλους τους διψήφιους περιττούς αριθμούς.

Στο τέλος αποθηκεύστε το πρόγραμμα στο φάκελό σας με το όνομα **python5.1**.

```
>>> %Run 5.1_odd.py
```

```
11  
13  
15  
17  
19  
21  
23  
25  
27  
29  
31  
33  
35  
37  
39  
41  
43  
45  
47  
49  
51  
53  
55
```

```
57  
59  
61  
63  
65  
67  
69  
71  
73  
75  
77  
79  
81  
83  
85  
87  
89  
91  
93  
95  
97  
99
```


Δραστηριότητα 5.3 – προπαίδεια

Να γράψετε πρόγραμμα που θα διαβάσει από τον χρήστη έναν ακέραιο αριθμό και θα εμφανίζει την προπαίδειά του. Θεωρήστε ότι η απάντηση του χρήστη θα είναι έγκυρη (ακέραιος αριθμός στο διάστημα [1-10]).

Στο τέλος αποθηκεύστε το πρόγραμμα στο φάκελό σας με το όνομα **python5.3**.

```
>>> %Run propedia.py
```

```
Διάλεξε αριθμό από το 1 έως το 10
```

```
7
```

```
1 x 7 = 7  
2 x 7 = 14  
3 x 7 = 21  
4 x 7 = 28  
5 x 7 = 35  
6 x 7 = 42  
7 x 7 = 49  
8 x 7 = 56  
9 x 7 = 63  
10 x 7 = 70
```

```
>>> %Run propedia.py
```

```
Διάλεξε αριθμό από το 1 έως το 10
```

```
3
```

```
1 x 3 = 3  
2 x 3 = 6  
3 x 3 = 9  
4 x 3 = 12  
5 x 3 = 15  
6 x 3 = 18  
7 x 3 = 21  
8 x 3 = 24  
9 x 3 = 27  
10 x 3 = 30
```

Λυμένο Παράδειγμα 2 – Υπολογισμός αθροίσματος ή μέσου όρου

Να γράψετε πρόγραμμα που θα διαβάζει τους 10 βαθμούς που πήρε ένας μαθητής και θα υπολογίζει και θα εμφανίζει τον μέσο όρο του.

Λύση 1η

```
1 grade1=int(input("Δώσε βαθμό για μάθημα 1\n"))
2 grade2=int(input("Δώσε βαθμό για μάθημα 2\n"))
3 grade3=int(input("Δώσε βαθμό για μάθημα 3\n"))
4 grade4=int(input("Δώσε βαθμό για μάθημα 4\n"))
5 grade5=int(input("Δώσε βαθμό για μάθημα 5\n"))
6 grade6=int(input("Δώσε βαθμό για μάθημα 6\n"))
7 grade7=int(input("Δώσε βαθμό για μάθημα 7\n"))
8 grade8=int(input("Δώσε βαθμό για μάθημα 8\n"))
9 grade9=int(input("Δώσε βαθμό για μάθημα 9\n"))
10 grade10=int(input("Δώσε βαθμό για μάθημα 10\n"))
11 mo = (grade1 + grade2 + grade3 + grade4 + grade5
12       + grade6 + grade7 + grade8 + grade9 + grade10)/10
13 print("Ο τελικός βαθμός σου είναι:", mo)
```

Παράδειγμα εκτέλεσης

```
>>> %Run example2_grade_no_iteration.py
```

```
Δώσε βαθμό για μάθημα 1
20
Δώσε βαθμό για μάθημα 2
18
Δώσε βαθμό για μάθημα 3
19
Δώσε βαθμό για μάθημα 4
17
Δώσε βαθμό για μάθημα 5
17
Δώσε βαθμό για μάθημα 6
18
Δώσε βαθμό για μάθημα 7
20
Δώσε βαθμό για μάθημα 8
19
Δώσε βαθμό για μάθημα 9
20
Δώσε βαθμό για μάθημα 10
20
Ο τελικός βαθμός σου είναι: 18.8
```

Λυμένο Παράδειγμα 2 – Υπολογισμός αθροίσματος ή μέσου όρου

Στην προηγούμενη λύση παρατηρούμε ότι ένα μεγάλο κομμάτι του κώδικα επαναλαμβάνεται **10 φορές**. Αν γίνει χρήση της δομής επανάληψης **for** ο κώδικας γίνεται:

Λύση 2η

```
1 total = 0
2 for i in range(1, 11):
3     print("Δώσε βαθμό για μάθημα", i, "\n")
4     grade=int(input())
5     total = total + grade
6     print("_____ \n")
7 mo = total/10
8 print("Ο τελικός βαθμός είναι:", mo)
```

Εξήγηση λύσης

1. Αρχικοποιώ μεταβλητή **total** για το άθροισμα όλων των βαθμών, στο 0
2. Χρησιμοποιώ for που να εκτελείται 10 φορές, όσες και οι βαθμοί
3. Σε κάθε μια από τις 10 επαναλήψεις εμφανίζω κατάλληλο μήνυμα
4. Διαβάζω τον επόμενο βαθμό με τη χρήση της **input**
5. Αυξάνω την μεταβλητή total κατά τον βαθμό που διάβασα (**grade**)
6. Τυπώνω ένα κατάλληλο διαχωριστικό (_____)
7. Με το που τελειώσει η for (**προσοχή στη στοίχιση**), υπολογίζω τον μέσο όρο διαιρώντας το άθροισμα (total) με το 10
8. Εκτυπώνω το τελικό αποτέλεσμα

Δραστηριότητα 5.4 – εκδρομή

Το Μουσικό Γυμνάσιο Παλλήνης έχει συνολικά **12 τμήματα** και **300 μαθητές**. Προκειμένου να γίνει μια ημερήσια εκδρομή θα πρέπει το ποσοστό των μαθητών που συμμετέχουν να είναι **τουλάχιστον το 70%** του συνόλου των μαθητών.

Να γράψετε πρόγραμμα που:

- θα διαβάζει για κάθε τμήμα πόσα παιδιά του τμήματος ενδιαφέρονται για συμμετοχή στην εκδρομή (**plithos** – ακέραιος)
- στο τέλος θα εμφανίζει το σύνολο των μαθητών του σχολείου που ενδιαφέρονται, το ποσοστό τους στο σύνολο του σχολείου και ένα μήνυμα για το αν τελικά θα πραγματοποιηθεί η εκδρομή ή όχι.

Αποθηκεύστε το πρόγραμμα στο φάκελό σας με το όνομα **python5.4**.

Παράδειγμα εκτέλεσης

```
>>> %Run -c $EDITOR_CONTENT
```

```
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 1 ;
20
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 2 ;
16
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 3 ;
14
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 4 ;
20
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 5 ;
20
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 6 ;
18
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 7 ;
15
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 8 ;
16
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 9 ;
18
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 10 ;
20
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 11 ;
22
Πόσοι μαθητές ενδιαφέρονται από το τμήμα 12 ;
21
Σύνολο μαθητών που ενδιαφέρονται: 220
Ποσοστό μαθητών που ενδιαφέρονται: 73.33333333333333 %
Η εκδρομή θα γίνει!
```

Δραστηριότητα 5.5 – πρωτάθλημα

Όπως έχουμε ξαναδεί, σε έναν αγώνα ποδοσφαίρου η **νίκη** δίνει **3 βαθμούς**, η **ισοπαλία** **1** και η **ήττα** **0**. Ένα μικρό περιφερειακό πρωτάθλημα έχει **8 αγωνιστικές**. Να γράψετε πρόγραμμα που:

- για κάθε αγωνιστική θα διαβάζει τα γκολ που πέτυχε (**goal1**) και τα γκολ που δέχθηκε η ομάδα (**goal2**)
- θα εμφανίζει τους πόντους που πήρε η ομάδα σε αυτή την αγωνιστική
- θα εμφανίζει τους συνολικούς πόντους που έχει μέχρι εκείνη τη στιγμή η ομάδα στο πρωτάθλημα
- στο τέλος θα ελέγχει και θα εμφανίζει μήνυμα για το αν η ομάδα πέρασε στα play off. Για να συμβεί αυτό πρέπει να συγκεντρώσει τουλάχιστον 15 βαθμούς.
- Αποθηκεύστε το πρόγραμμα στο φάκελό σας με το όνομα **python5.5**.

Παράδειγμα εκτέλεσης

```
>>> %Run -c $EDITOR_CONTENT
```

```
Αγωνιστική 1
Πόσα γκολ έβαλε η ομάδα;
3
Πόσα γκολ δέχθηκε η ομάδα;
2
Πόντοι αγωνιστικής: 3
Συνολικοί πόντοι: 3
```

```
Αγωνιστική 2
Πόσα γκολ έβαλε η ομάδα;
0
Πόσα γκολ δέχθηκε η ομάδα;
0
Πόντοι αγωνιστικής: 1
Συνολικοί πόντοι: 4
```

```
Αγωνιστική 3
Πόσα γκολ έβαλε η ομάδα;
1
Πόσα γκολ δέχθηκε η ομάδα;
1
Πόντοι αγωνιστικής: 1
Συνολικοί πόντοι: 5
```

```
Αγωνιστική 4
Πόσα γκολ έβαλε η ομάδα;
2
Πόσα γκολ δέχθηκε η ομάδα;
0
Πόντοι αγωνιστικής: 3
Συνολικοί πόντοι: 8
```

```
Αγωνιστική 5
Πόσα γκολ έβαλε η ομάδα;
3
Πόσα γκολ δέχθηκε η ομάδα;
2
Πόντοι αγωνιστικής: 3
Συνολικοί πόντοι: 11
```

```
Αγωνιστική 6
Πόσα γκολ έβαλε η ομάδα;
0
Πόσα γκολ δέχθηκε η ομάδα;
1
Πόντοι αγωνιστικής: 0
Συνολικοί πόντοι: 11
```

```
Αγωνιστική 7
Πόσα γκολ έβαλε η ομάδα;
1
Πόσα γκολ δέχθηκε η ομάδα;
0
Πόντοι αγωνιστικής: 3
Συνολικοί πόντοι: 14
```

```
Αγωνιστική 8
Πόσα γκολ έβαλε η ομάδα;
1
Πόσα γκολ δέχθηκε η ομάδα;
1
Πόντοι αγωνιστικής: 1
Συνολικοί πόντοι: 15
```

Πρόκριση στα playoff!

Λυμένο Παράδειγμα 3 – Υπολογισμός πλήθους

Να γράψετε πρόγραμμα που θα διαβάσει την ηλικία 20 επισκεπτών σε κάποιο εμπορικό κέντρο και στο τέλος θα εμφανίζει το πλήθος και το ποσοστό των επισκεπτών αυτών που είναι ανήλικοι.

Παράδειγμα εκτέλεσης

```
>>> %Run example3_plithos.py
```

```
Ηλικία επισκέπτη 1
25
Ηλικία επισκέπτη 2
22
Ηλικία επισκέπτη 3
17
Ηλικία επισκέπτη 4
16
Ηλικία επισκέπτη 5
17
Ηλικία επισκέπτη 6
56
Ηλικία επισκέπτη 7
45
Ηλικία επισκέπτη 8
38
Ηλικία επισκέπτη 9
47
Ηλικία επισκέπτη 10
14
```

```
Ηλικία επισκέπτη 11
16
Ηλικία επισκέπτη 12
19
Ηλικία επισκέπτη 13
15
Ηλικία επισκέπτη 14
27
Ηλικία επισκέπτη 15
36
Ηλικία επισκέπτη 16
67
Ηλικία επισκέπτη 17
64
Ηλικία επισκέπτη 18
32
Ηλικία επισκέπτη 19
21
Ηλικία επισκέπτη 20
15
```

```
Το πλήθος των ανήλικων είναι: 7
Το ποσοστό των ανήλικων είναι: 35.0 %
```

Λυμένο Παράδειγμα 3 – Υπολογισμός πλήθους



Εξήγηση λύσης

Λύση

```
1 plithos=0
2 for x in range (1,21,1):
3     print("Ηλικία επισκέπτη", x)
4     age=int(input())
5     if age<18:
6         plithos=plithos+1
7 pososto=plithos/20*100
8 print("_____")
9 print("Το πλήθος των ανήλικων είναι:",plithos)
10 print("Το ποσοστό των ανήλικων είναι:",pososto,"%")
```

1. Αρχικοποιώ μεταβλητή **plithos** για το πλήθος των ανήλικων, στο 0
2. Χρησιμοποιώ for που να εκτελείται 20 φορές, όσες και οι επισκέπτες που θα ερωτηθούν
3. Σε κάθε μια από τις 20 επαναλήψεις εμφανίζω κατάλληλο μήνυμα
4. Διαβάζω την ηλικία του επισκέπτη με τη χρήση της **input**
5. Με τη χρήση της **if** ελέγχω αν η ηλικία είναι μικρότερη του 18
6. Σε περίπτωση που ισχύει αυτό αυξάνω το πλήθος κατά 1
7. Με το που τελειώσει η for (**προσοχή στη στοίχιση**), υπολογίζω το ποσοστό των ανηλίκων
8. Τυπώνω ένα κατάλληλο διαχωριστικό (_____)
9. Εκτυπώνω το πλήθος των ανήλικων
10. Εκτυπώνω το ποσοστό των ανήλικων.

Δραστηριότητα 5.6 – προορισμός εκδρομής

Οι **25** μαθητές ενός τμήματος καλούνται να επιλέξουν ανάμεσα σε δύο υποψήφιους προορισμούς για ημερήσια εκδρομή. Με την τιμή 1 επιλέγουν τη Χαλκίδα ενώ με την τιμή 2 το Ναύπλιο. Να γράψετε πρόγραμμα που:

- θα διαβάσει για κάθε μαθητή την επιλογή του (**choice** – ακέραιος). Αν δώσει απάντηση διαφορετική από το 0 ή το 1 θα αγνοείται.
- στο τέλος θα εμφανίζει το πλήθος και ποσοστό των μαθητών που επέλεξαν Χαλκίδα και Ναύπλιο καθώς και σε ποιον προορισμό θα γίνει τελικά η εκδρομή. Αν τα δύο ποσοστά είναι ίσα θα εμφανίζεται μήνυμα για επανάληψη της ψηφοφορίας.

Αποθηκεύστε το πρόγραμμα στο φάκελό σας με το όνομα **python5.6**.

Παράδειγμα εκτέλεσης

```
>>> %Run 5.6_proorismos.py
```

```
Επιλογή προορισμού μαθητή 1
Δώσε 1 για Χαλκίδα, 2 για Ναύπλιο
1
Επιλογή προορισμού μαθητή 2
Δώσε 1 για Χαλκίδα, 2 για Ναύπλιο
1
Επιλογή προορισμού μαθητή 3
Δώσε 1 για Χαλκίδα, 2 για Ναύπλιο
2
Επιλογή προορισμού μαθητή 4
Δώσε 1 για Χαλκίδα, 2 για Ναύπλιο
2
Επιλογή προορισμού μαθητή 5
Δώσε 1 για Χαλκίδα, 2 για Ναύπλιο
1
Επιλογή προορισμού μαθητή 6
Δώσε 1 για Χαλκίδα, 2 για Ναύπλιο
1
```

...

```
Επιλογή προορισμού μαθητή 24
Δώσε 1 για Χαλκίδα, 2 για Ναύπλιο
2
Επιλογή προορισμού μαθητή 25
Δώσε 1 για Χαλκίδα, 2 για Ναύπλιο
2
```

```
*****
Στη Χαλκίδα θέλουν να πάνε: 15 δηλαδή το 60.0 %
Στο Ναύπλιο θέλουν να πάνε: 10 δηλαδή το 40.0 %
Η εκδρομή θα γίνει στη Χαλκίδα!
```



While

Εκτός από τη **for** που μάθαμε, υπάρχει και η **while** που υλοποιεί τη δομή της επανάληψης.

Σύνταξη:

```
x = start  
while x < stop:  
    ομάδα εντολών 1  
    x = x + step
```

```
for x in range(start, stop, step):  
    ομάδα εντολών 1
```

While - γνωστός αριθμός επαναλήψεων

Παρακάτω βλέπουμε δυο ισοδύναμους κώδικες με for και while που δημιουργούν ένα τρίγωνο από *, όπως αυτό της διπλανής εικόνας.

Κώδικας με for

```
1 for x in range(1,11,1):
2     print(x*"*)
```

Κώδικας με while

```
1 x=1
2 while x<11:
3     print(x*"*)
4     x = x + 1
```

Εξήγηση κώδικα με while

1. Αρχικοποιούμε το x στο 1
2. Μέσα στη while βάζουμε τη συνθήκη x<11
3. Εκτυπώνουμε x πλήθος από *
4. Αυξάνουμε το x κατά 1

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

Στη for όλες οι πληροφορίες είναι μαζεμένες σε μια γραμμή και αυτό την κάνει πιο βολική όταν ο αριθμός των επαναλήψεων είναι γνωστός. Έτσι συνήθως την προτιμούμε σε σχέση με τη while.

While - άγνωστος αριθμός επαναλήψεων

Λυμένο Παράδειγμα 4



Η while είναι πιο γενική και σε αντίθεση με τη for, μπορεί να χρησιμοποιηθεί και όταν ο **αριθμός των επαναλήψεων είναι άγνωστος**.

Ας δούμε ένα παράδειγμα:

Έστω ότι έχετε στη διάθεσή σας το ποσό των 1000 ευρώ και αρχίζετε να αγοράζετε διάφορα πράγματα. Γράψτε πρόγραμμα που:

- Για κάθε αντικείμενο που θέλετε να αγοράσετε να διαβάζει την τιμή του.
- Αν τα λεφτά που έχετε επαρκούν να εμφανίζει το μήνυμα «Η αγορά έγινε» και να εμφανίζει το τρέχον υπόλοιπο.
- Σε αντίθετη περίπτωση να εμφανίζει το μήνυμα «Τα χρήματα δεν επαρκούν»
- Η διαδικασία θα ολοκληρώνεται με την πρώτη αδυναμία αγοράς αντικειμένου.
- Στο τέλος το πρόγραμμα να εμφανίζει πόσα αντικείμενα αγοράστηκαν και πόσα λεφτά ξοδεύτηκαν συνολικά.

Παράδειγμα εκτέλεσης

```
>>> %Run -c $EDITOR_CONTENT
Πόσο κοστίζει το αντικείμενο που θέλετε να αγοράσετε;250
Η αγορά έγινε
Διαθέσιμο ποσό: 750

Πόσο κοστίζει το αντικείμενο που θέλετε να αγοράσετε;300
Η αγορά έγινε
Διαθέσιμο ποσό: 450

Πόσο κοστίζει το αντικείμενο που θέλετε να αγοράσετε;400
Η αγορά έγινε
Διαθέσιμο ποσό: 50

Πόσο κοστίζει το αντικείμενο που θέλετε να αγοράσετε;100
Τα χρήματα δεν επαρκούν
*****
Συνολικά αγοράστηκαν: 3 αντικείμενα
και ξοδεύτηκαν: 950 Ευρώ
```

While - άγνωστος αριθμός επαναλήψεων

Λυμένο Παράδειγμα 4



```
1 money=1000
2 bought = 0
3 spent = 0
4 price=int(input("Πόσο κοστίζει το αντικείμενο που θέλετε να αγοράσετε;"))
5 while money-price>=0:
6     money=money-price
7     bought=bought+1
8     spent=spent+price
9     print ("Η αγορά έγινε")
10    print ("Διαθέσιμο ποσό:", money)
11    print("_____")
12    price=int(input("Πόσο κοστίζει το αντικείμενο που θέλετε να αγοράσετε;"))
13 print ("Τα χρήματα δεν επαρκούν")
14 print ("*****")
15 print ("Συνολικά αγοράστηκαν:", bought, "αντικείμενα")
16 print ("και ξοδεύτηκαν:", spent, "Ευρώ")
```

Λύση

Εξήγηση λύσης

- 1-3. Αρχικοποιώ το διαθέσιμο ποσό (money) στο 1000, τα αγορασμένα αντικείμενα (bought) στο 0 και τα χρήματα που ξοδεύτηκαν (spent) στο 0
4. Διαβάζω πόσο κοστίζει το αντικείμενο που θέλω να αγοράσω (price)
5. Χρησιμοποιώ τη while για την επανάληψη με συνθήκη $money-price \geq 0$ (όσο τα λεφτά που έχω αρκούν για την επόμενη αγορά)
- 6-8. Ενημερώνω κατάλληλα τις τιμές των μεταβλητών money, bought και spent
- 9-11. Εμφανίζω τα απαραίτητα μηνύματα
12. Διαβάζω την τιμή του επόμενου αντικειμένου που θέλω να αγοράσω.
- 13-16. Εμφανίζω μήνυμα ολοκλήρωσης της διαδικασίας και τα κατάλληλα συνολικά στατιστικά.

Δραστηριότητα 5.7 – ασανσέρ

Ένα ασανσέρ σε κάποιο ξενοδοχείο έχει μέγιστη χωρητικότητα **10 ατόμων** ή **800 συνολικά κιλών**. Να γράψετε πρόγραμμα που:

- Για κάθε άτομο που θέλει να μπει στο ασανσέρ να μετράει το βάρος του (σε κιλά) και να ελέγχει αν μπορεί να μπει με βάση τους δύο παραπάνω περιορισμούς. Εφόσον η απάντηση είναι θετική να εμφανίζει το μήνυμα «**Επιτρεπτή είσοδος**» και θα υπολογίζει και θα εμφανίζει πόσα άτομα χωράνε ακόμα στο ασανσέρ και πόσα τα διαθέσιμα επιτρεπτά κιλά .
- Θα τερματίζει τον έλεγχο εισόδου στο ασανσέρ, όταν έχουν μπει ήδη 10 άτομα ή όταν το βάρος του τελευταίου που θέλει να μπει στο ασανσέρ ξεπερνά το διαθέσιμο επιτρεπτό βάρος. Στο τέλος θα εμφανίζει πόσα άτομα επιβιβάστηκαν στο ασανσέρ και το συνολικό βάρος τους.

Αποθηκεύστε το πρόγραμμα στο φάκελό σας με το όνομα **python5.7**.

Παράδειγμα εκτέλεσης

```
>>> %Run '5.7_hard disc.py'
```

```
Βάρος ατόμου;
76
Επιτρεπτή είσοδος
Χωράνε ακόμα 9 άτομα ή 724 κιλά
```

```
Βάρος ατόμου;
83
Επιτρεπτή είσοδος
Χωράνε ακόμα 8 άτομα ή 641 κιλά
```

```
Βάρος ατόμου;
112
Επιτρεπτή είσοδος
Χωράνε ακόμα 7 άτομα ή 529 κιλά
```

...

```
Βάρος ατόμου;
100
Επιτρεπτή είσοδος
Χωράνε ακόμα 2 άτομα ή 105 κιλά
```

```
Βάρος ατόμου;
90
Επιτρεπτή είσοδος
Χωράνε ακόμα 1 άτομα ή 15 κιλά
```

```
Βάρος ατόμου;
78
Μη επιτρεπτή είσοδος
*****
Συνολικά εισήλθαν: 9 άτομα
και είχαν βάρος: 785 κιλά
```



Δραστηριότητα 5.8 – PIN κινητού

Κατά την επανεκκίνηση κινητού ζητείται από τον χρήστη να εισάγει ένα τετραψήφιο **PIN** προκειμένου να μπορεί να το χρησιμοποιήσει. Να γράψετε πρόγραμμα που:

- Θα θεωρεί δεδομένο το πραγματικό τετραψήφιο PIN (θα έχει την τιμή 1351) .
- Θα ζητάει από τον χρήστη να δώσει το PIN και αν αυτός δώσει το σωστό θα τον καλωσορίζει στο κινητό, ενώ αν δώσει λάθος θα του ζητάει εκ νέου να δώσει το PIN
- Η διαδικασία μπορεί να επαναληφθεί μέχρι 3 το πολύ φορές. Μετά από 3 αποτυχημένες προσπάθειες εμφανίζεται κατάλληλο μήνυμα κλειδώματος του κινητού

Αποθηκεύστε το πρόγραμμα στο φάκελό σας με το όνομα **python5.8**.

Παραδείγματα εκτέλεσης

```
>>> %Run 5.8_pin.py
Δώσε PIN κινητού
1351
Καλωσήρθατε!
```

```
>>> %Run 5.8_pin.py
Δώσε PIN κινητού
1531
Δώσε PIN κινητού
1353
Δώσε PIN κινητού
1351
Καλωσήρθατε!
```

```
>>> %Run 5.8_pin.py
Δώσε PIN κινητού
1353
Δώσε PIN κινητού
3513
Δώσε PIN κινητού
1153
Κινητό κλειδωμένο (3 αποτυχημένες προσπάθειες)
```