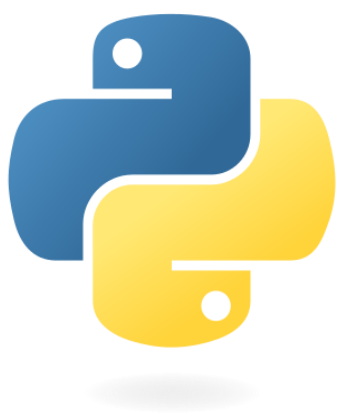


Προγραμματισμός στη γλώσσα



python

Περιεχόμενα

A. Τύποι Δεδομένων	3
B. Μεταβλητές	5
Γ. Τελεστές, Σχόλια.....	6
Γ.1 Αριθμητικοί Τελεστές.....	6
Γ.2 Λογικοί Τελεστές	8
Γ.3 Σχεσιακοί Τελεστές.....	8
Γ.4 Σχόλια	9
Δ. Είσοδος δεδομένων – Έξοδος αποτελεσμάτων.....	10
Δ.1. Είσοδος δεδομένων από το πληκτρολόγιο	10
Δ.2. Έξοδος αποτελεσμάτων	10
E. Αλγοριθμικές Δομές	11
E.1. Δομή Ακολουθίας ή Ακολουθιακή Δομή	11
E.2 Δομή Επιλογής	13
Απλή Επιλογή	13
Σύνθετη Επιλογή	14
Πολλαπλή επιλογή.....	16
Εμφωλευμένη επιλογή	17
E.3 Δομή Επανάληψης ή Επαναληπτική Δομή	18
Η εντολή for	18
Η εντολή while	20
ΣΤ. Συναρτήσεις	21

A. Τύποι Δεδομένων



Οι βασικοί τύποι δεδομένων είναι:

- **ακέραιος αριθμός (int)**, π.χ. -6, 0, 1, 587
- **πραγματικός αριθμός ή αριθμός κινητής υποδιαστολής (float)**, π.χ. 3.14159, 0.6, 1.333
- **αλφαριθμητικό ή συμβολοσειρά (str)**, π.χ. "2ο Γυμνάσιο Μηχανιώνας", «Hello world". Το αλφαριθμητικό ή η συμβολοσειρά είναι μια ακολουθία από χαρακτήρες (ψηφία, γράμματα, σύμβολα) και βρίσκεται μέσα σε εισαγωγικά μονά ή διπλά.
- **λογικός (bool)** π.χ. True, False. Ο λογικός τύπος (boolean) έχει μόνο δύο τιμές: την τιμή True (Αληθής) και τη τιμή False (Ψευδής)
- **λίστα (list)** π.χ. ['μήλο', 'αχλάδι', 'πορτοκάλι'], ['ΠΕΤΡΑ', 'ΨΑΛΙΔΙ', 'ΧΑΡΤΙ'], [1, 2, 3]. Η λίστα είναι μια ακολουθία από αντικείμενα οποιουδήποτε τύπου. Είναι δυναμική δομή, δηλαδή μπορούμε να προσθέσουμε ή να αφαιρέσουμε στοιχεία αυξομειώνοντας το μέγεθός της.

Για να τυπώσουμε τιμές, μπορούμε να χρησιμοποιήσουμε την εντολή **print**.



Εφαρμογή A.1. Γράψε και εκτέλεσε τις παρακάτω εντολές:

```
print(3)
print(3.14)
print('hello world')
```

```
>>> type(3)
<class 'int'>
>>> type(3.14)
<class 'float'>
>>> type("Hello world")
<class 'str'>
```

Αν δεν γνωρίζουμε τον τύπο στον οποίο ανήκει μία τιμή, μπορούμε να χρησιμοποιήσουμε την εντολή **type**.

```
>>> type(False)
<class 'bool'>
>>> type(True)
<class 'bool'>
>>> type([1,2,3])
<class 'list'>
>>> type(['apple', 'peach', 'orange'])
<class 'list'>
```



Εφαρμογή Α.2. Γράψε και εκτέλεσε τις παρακάτω εντολές. Τι παρατηρείς;

```
type('3')
type('3.14')
```

Οι τιμές '3' και '3.14' είναι μέσα σε εισαγωγικά. Άρα, είναι συμβολοσειρά (*str*).



Εφαρμογή Α.3. Γράψε και εκτέλεσε τις παρακάτω εντολές. Τι παρατηρείς;

```
print(2,5)
print(2.5)
```

Η 1^η εντολή τυπώνει τους αριθμούς 2 και 5. Η 2^η εντολή τον δεκαδικό αριθμό 2,5.



Στον προγραμματισμό για την υποδιαστολή χρησιμοποιείται η τελεία «.» και όχι το κόμμα.

```
>>> int(3.9)
3
>>> float(5)
5.0
>>> 5*int('28')
140
>>> 5*str(12)
'1212121212'
```

Συναρτήσεις μετατροπής από έναν τύπο σε ένα άλλο:

- Η συνάρτηση **int** μετατρέπει πραγματικούς αριθμούς και αλφαριθμητικά σε ακέραιους
- Η συνάρτηση **float** μετατρέπει ακέραιους αριθμούς και αλφαριθμητικά σε πραγματικούς
- Η συνάρτηση **str** μετατρέπει αριθμούς σε αλφαριθμητικά

```
>>> len('python')
6
>>> len('1024')
4
>>> len(str(1024))
4
```

Η συνάρτηση **len** επιστρέφει το πλήθος των χαρακτήρων (μήκος) ενός αλφαριθμητικού

```
>>> a='computer'
>>> a[0]
'c'
>>> a[1]
'o'
>>> a[2]
'm'
>>> a[3]
'p'
```

```
>>> 'computer'[2:4]
'mp'
>>> 'computer'[3:6]
'put'
```

Μελέτησε τις διπλανές εντολές και προσπάθησε να τις κατανοήσεις!

Ποιος χαρακτήρας είναι το `a[4]`; Τι θα επιστρέψει η έκφραση `'computer'[4,7]`;

B. Μεταβλητές



Μια **μεταβλητή (variable)** είναι ένα όνομα που αναφέρεται σε μία τιμή. Η τιμή αυτή μπορεί να αλλάζει κατά την εκτέλεση ενός προγράμματος. Η μεταβλητή είναι τμήμα της μνήμης του υπολογιστή μας όπου μπορούμε να αποθηκεύσουμε μόνο μία τιμή κάθε φορά.

Η εντολή εκχώρησης (π.χ. `n = 3`) δημιουργεί νέα μεταβλητή (τη μεταβλητή `n`) και της αποδίδει μία τιμή (την τιμή 3). Αποτελείται από 3 μέρη:

1^ο μέρος: το όνομα της μεταβλητής

2^ο μέρος: το =

3^ο μέρος: μία τιμή, μία άλλη μεταβλητή, μια έκφραση

```
>>> day='Sunday'
>>> print(day)
Sunday
>>> type(day)
<class 'str'>
```

```
>>> a=6
>>> print(a)
6
>>> type(a)
<class 'int'>
```

```
>>> pi=3.14159
>>> print(pi)
3.14159
>>> type(pi)
<class 'float'>
```

```
>>> b=3+5
>>> print(b)
8
```

```
>>> a=10
>>> b=a
>>> print(b)
10
```

```
>>> a=2
>>> b=5
>>> a=b
>>> print(a)
5
>>> print(b)
5
```

```
>>> n=10
>>> n=n+1
>>> print(n)
11
```



Ονόματα μεταβλητών – Κανόνες

- Τα ονόματα των μεταβλητών μπορούν να περιέχουν γράμματα και αριθμούς, αλλά πρέπει να ξεκινούν με ένα γράμμα ή τον χαρακτήρα `_`.
- Τα πεζά διακρίνονται από τα κεφαλαία γράμματα. Π.χ. οι `ram`, `Ram`, `RAm`, `RAM`, `rAm` είναι διαφορετικές μεταβλητές.
- Δεν επιτρέπονται το κενό, τα εισαγωγικά, η τελεία, το κόμμα και άλλοι παρόμοιοι χαρακτήρες. Επιτρέπεται μόνο ο χαρακτήρας `_`.

Γ. Τελεστές, Σχόλια

Γ.1 Αριθμητικοί Τελεστές

Τελεστής	Εξήγηση	Παραδείγματα
+	Πρόσθεση αριθμών ή ένωση αλφαριθμητικών	<pre>>>> 2+6 8 >>> 'α'+'β' 'αβ'</pre>
-	Αφαίρεση αριθμών	<pre>>>> 10-4 6 >>> 0-12 -12</pre>
*	Γινόμενο αριθμών ή επανάληψη ενός αλφαριθμητικού	<pre>>>> 2*5 10 >>> 2*3*6 36 >>> 4*'ha' 'hahahaha'</pre>
/	Διαίρεση 2 αριθμών. Το αποτέλεσμα είναι πάντα πραγματικός αριθμός float	<pre>>>> 6/4 1.5 >>> 8/2 4.0 >>> 4/3 1.3333333333333333</pre>
**	Ύψωση αριθμού σε δύναμη	<pre>>>> 3**4 81 >>> 2**10 1024 >>> 5.0**3 125.0</pre>
//	Πηλίκo ακέραιας διαίρεσης	<pre>>>> 5//2 2 >>> 20.0//3 6.0</pre>

%	Υπόλοιπο ακέραιας διαίρεσης	<pre>>>> 5%2 1 >>> 20.0%3 2.0</pre>
---	-----------------------------	---



Εφαρμογή Γ.1. Με ποιο τρόπο μπορείς να κάνεις αντιμετάθεση τιμών 2 μεταβλητών (**swap**); Είναι σωστό το παρακάτω;

```
a=10
b=5
a=b
b=a
```

Μελέτησε τους παρακάτω τρόπους για την αντιμετάθεση 2 μεταβλητών:

1 ^{ος} τρόπος	2 ^{ος} τρόπος	3 ^{ος} τρόπος
<u>με βοηθητική μεταβλητή</u>	<u>χωρίς βοηθητική μεταβλητή</u>	<u>με την έκφραση</u> a, b = b, a
a = 5	a = 10	a = 10
b = 10	b = 5	b = 5
temp = a	a = a + b	a, b = b, a
a = b	b = a - b	
b = temp	a = a - b	



Η προτεραιότητα τέλεσης των πράξεων όπως και στα μαθηματικά είναι:

1. εκφράσεις μέσα σε παρενθέσεις
2. δυνάμεις
3. πολλαπλασιασμοί και διαιρέσεις
4. προσθέσεις και αφαιρέσεις



Εφαρμογή Γ.2. Γράψε και εκτέλεσε τις παρακάτω εκφράσεις. Τι παρατηρείς στην τελευταία έκφραση;

```
>>> 300+1*20
>>> (300+1)*20
>>> 2*-1
```

Γ.2 Λογικοί Τελεστές



Στα προγράμματά μας χρειάζεται συχνά να ελέγχουμε αν ισχύει (True) ή όχι (False) μία συνθήκη. Θυμήσου τη συνάρτηση IF από το υπολογιστικό φύλλο (Excel)! Στον προγραμματισμό χρησιμοποιείται η **λογική Boolean**. Στην Python ο λογικός τύπος είναι ο bool. Μία μεταβλητή μπορεί να δείχνει και σε μία τιμή τύπου bool. Π.χ.

```
>>> a=True
>>> type(a)
<class 'bool'>
>>> b=False
>>> type(b)
<class 'bool'>
```

Όταν θέλουμε να σχηματίσουμε σύνθετες συνθήκες χρησιμοποιούμε τους **λογικούς τελεστές**.

Άρνηση (Όχι)	not
Σύζευξη (ΚΑΙ)	and
Διάζευξη (Η)	or

Γ.3 Σχεσιακοί Τελεστές

Σε μία λογική έκφραση μπορούμε να έχουμε και **τελεστές σύγκρισης** ή **σχεσιακούς τελεστές**.

Μικρότερο από	<
Μικρότερο ή ίσο από	<=
Μεγαλύτερο από	>
Μεγαλύτερο ή ίσο από	>=
Ίσο με	==
Διάφορο από	!=



Εφαρμογή Γ.3. Γράψε και εκτέλεσε τις εκφράσεις:

```
>>> 1024==1024
True
>>> 1==2
False
>>> 10>7
True
>>> 10>=10
True
>>> 3<3
False
>>> 3<=3
True
```

```
>>> 10>9 and 10<12
True
>>> 10<9 and 9<15
False
>>> 10<9 or 9<15
True
>>> 10<11 or 10<12
True
>>> 10>11 or 10>12
False
```

```
>>> 1!=2
True
>>> not(10<=9)
True
>>> not(10>9)
False
```



Προσοχή:

Ο == είναι τελεστής ισότητας, ενώ ο = είναι τελεστής εκχώρησης!

```
>>> a=3
>>> print(a)
3
>>> b=3
>>> a==b
True
```

Γ.4 Σχόλια



Πολλές φορές για να γίνει κατανοητό ένα πρόγραμμα είναι χρήσιμο να προσθέσουμε σχόλια στα προγράμματα. Τα σχόλια στη Python αρχίζουν με τον χαρακτήρα #. Οτιδήποτε ακολουθεί ένα μετά το #, η Python το αγνοεί.

```
#Υπολογισμός εμβαδού τετραγώνου

a=5 #πλευρά τετραγώνου
E=a*a #υπολογισμός εμβαδού τετραγώνου
print('Εμβαδόν=',E) #εκτύπωση
```

Δ. Είσοδος δεδομένων – Έξοδος αποτελεσμάτων

Δ.1. Είσοδος δεδομένων από το πληκτρολόγιο



Σε πολλά προγράμματα απαιτείται η είσοδος δεδομένων από τον χρήστη. Η Python διαθέτει τη συνάρτηση **input**. Κάθε φορά που καλείται η συνάρτηση **input** το πρόγραμμα σταματάει και περιμένει από τον χρήστη να πληκτρολογήσει κάτι. Μόλις ο χρήστης πληκτρολογήσει και πατήσει enter, τότε το πρόγραμμα συνεχίζει και η συνάρτηση **input** επιστρέφει την είσοδο σαν αλφαριθμητικό (**str**). Καλό θα είναι να εμφανίζεται ένα μήνυμα το οποίο να ενημερώνει τον χρήστη για το τι αναμένεται από αυτόν να πληκτρολογήσει.

```
name=input()
```

```
name=input('Πώς σε λένε;')
```

Για παράδειγμα, στην 1^η περίπτωση ο χρήστης δεν θα γνωρίζει τι να πληκτρολογήσει, ενώ στη 2^η θα εμφανιστεί το μήνυμα 'Πώς σε λένε;' και θα καταλάβει ώστε να πληκτρολογήσει το όνομά του.

```
age=int(input('Πόσο χρονών είσαι;'))
```

```
number=float(input('Δώσε έναν αριθμό:'))
```

Επειδή, η συνάρτηση **input** επιστρέφει πάντα αλφαριθμητικό, ανάλογα με τον τύπο των δεδομένων χρειάζεται η κατάλληλη μετατροπή.

Δ.2. Έξοδος αποτελεσμάτων



Η εντολή **print** εμφανίζει μηνύματα και περιεχόμενο μεταβλητών στην οθόνη και μπορούμε να τη χρησιμοποιήσουμε όταν θέλουμε να εμφανίσουμε τα αποτελέσματα ενός προγράμματος.



Εφαρμογή Δ.1. Γράψε και εκτέλεσε τα παρακάτω προγράμματα. Τι παρατηρείς;

```
number=input('Δώσε έναν αριθμό:')  
print(3*number)
```

```
number=int(input('Δώσε έναν αριθμό:'))  
print(3*number)
```

Ε. Αλγοριθμικές Δομές

Ε.1. Δομή Ακολουθίας ή Ακολουθιακή Δομή



Η **Δομή Ακολουθίας** αναφέρεται στις εντολές που εκτελούνται με τη σειρά, δηλαδή **ακολουθιακά** σε ένα πρόγραμμα.



Άσκηση 1^η. Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει το μήκος της πλευράς (ακέραιος αριθμός) ενός τετραγώνου και στη συνέχεια θα υπολογίζει και θα εμφανίζει το Εμβαδόν και την Περίμετρό του.



Άσκηση 2^η. Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει το μήκος των 2 πλευρών (ακέραιοι αριθμοί) ενός ορθογωνίου και στη συνέχεια θα υπολογίζει και θα εμφανίζει το Εμβαδόν και την Περίμετρό του.



Άσκηση 3^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει το μήκος της βάσης και το μήκος του ύψους (ακέραιοι αριθμοί) ενός τριγώνου και στη συνέχεια θα υπολογίζει και θα εμφανίζει το Εμβαδόν του.



Άσκηση 4^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει την ακτίνα (ακέραιος αριθμός) ενός κύκλου και στη συνέχεια θα υπολογίζει και θα εμφανίζει το Εμβαδόν και το Μήκος του κύκλου.



Άσκηση 5^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει την ηλικία ενός χρήστη και στη συνέχεια θα υπολογίζει και θα εμφανίζει τους μέγιστους καρδιακούς παλμούς ($ΜΚΠ = 220 - \text{ηλικία}$).



Άσκηση 6^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει 2 ακέραιους αριθμούς και στη συνέχεια θα υπολογίζει και θα εμφανίζει το άθροισμα, τη διαφορά, το γινόμενο και το πηλίκο των 2 αριθμών.



Άσκηση 7^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάζει τη θερμοκρασία (ακέραιος αριθμός) σε βαθμούς Κελσίου και θα τη μετατρέπει σε βαθμούς [Φαρενάιτ](#).
$$F = C * 1.8 + 32$$



Άσκηση 8^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάζει τη θερμοκρασία (ακέραιος αριθμός) σε βαθμούς Φαρενάιτ και θα τη μετατρέπει σε βαθμούς [Κελσίου](#).
$$C = (F - 32) / 1.8$$



Άσκηση 9^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάζει ένα αριθμό x και θα υπολογίζει την [συνάρτηση](#) $f(x) = x^2 + 2x + 1$.

E.2 Δομή Επιλογής



Η **Δομή Επιλογής** περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (**True** ή **False**) και την εκτέλεση κάποιας ενέργειας με βάση την τιμή της λογικής αυτής συνθήκης. Υπάρχουν

τέσσερις μορφές επιλογής:

- Απλή επιλογή
- Σύνθετη Επιλογή
- Πολλαπλή Επιλογή
- Εμφωλευμένη Επιλογή

Απλή Επιλογή



Την **Απλή Επιλογή** τη χρησιμοποιούμε όταν θέλουμε να εκτελεστεί μια σειρά εντολών, όταν ισχύει μια συνθήκη. Αν ισχύει η συνθήκη, δηλαδή είναι **True**, τότε μόνο εκτελούνται οι εντολές.

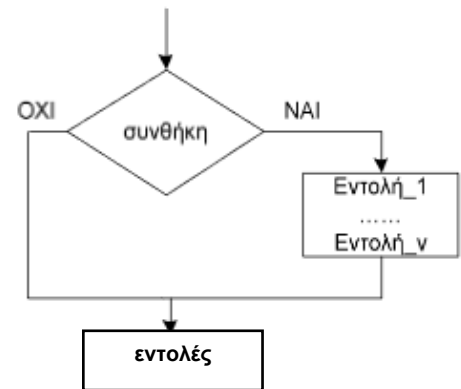
if <συνθήκη>:

 εντολή_1

 ...

 εντολή_v

εντολές



Άσκηση 10^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάζει έναν κωδικό. Αν ο κωδικός είναι το secret, θα εμφανίζει το μήνυμα «Επιτυχής Σύνδεση».



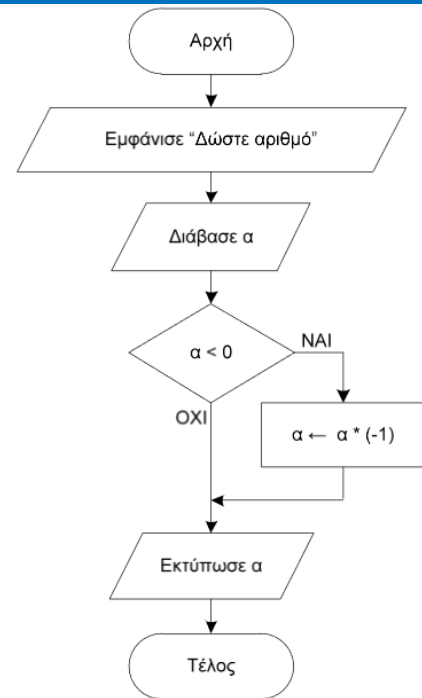
Άσκηση 11^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάζει, θα ελέγχει την ηλικία ενός χρήστη και θα εμφανίζει το μήνυμα «Είσαι ανήλικος/η».



Άσκηση 12^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάζει και θα ελέγχει τη θερμοκρασία. Αν η θερμοκρασία είναι μικρότερη των 15 βαθμών Κελσίου θα εμφανίζει τα μηνύματα «Κάνει κρύο σήμερα.» και «Ζακέτα να πάρεις!». Σε κάθε περίπτωση να εμφανίζει μήνυμα «Ωρα να πας σχολείο τώρα.»



Άσκηση 13^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει έναν πραγματικό αριθμό και στη συνέχεια θα εμφανίζει την απόλυτη τιμή του.



Άσκηση 14^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει 3 ακέραιους αριθμούς και θα εμφανίζει τον μεγαλύτερο.



Άσκηση 15^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει 3 ακέραιους αριθμούς και θα εμφανίζει τον μικρότερο.

Σύνθετη Επιλογή



Τη δομή της **Σύνθετης Επιλογής** τη χρησιμοποιούμε όταν θέλουμε να εκτελεστούν κάποιες εντολές, όταν η συνθήκη ισχύει (True) και κάποιες άλλες εντολές όταν η συνθήκη δεν ισχύει (False).

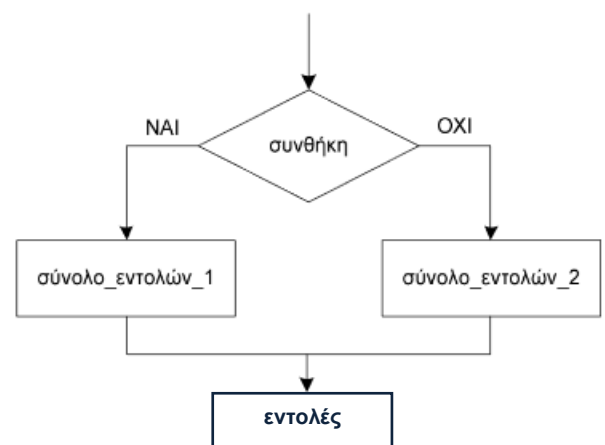
if <συνθήκη>:

σύνολο_εντολών1

else:

σύνολο_εντολών2

εντολές





Άσκηση 16^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει το [έτος γέννησης](#) ενός χρήστη, θα υπολογίζει και θα εμφανίζει την ηλικία του και θα ελέγχει και θα εμφανίζει αν είναι ενήλικος ή ανήλικος.



Άσκηση 17^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει και θα ελέγχει τον [κωδικό](#) εισόδου ενός χρήστη στο λογαριασμό του και θα εμφανίζει ανάλογο μήνυμα.



Άσκηση 18^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει έναν ακέραιο αριθμό και θα εμφανίζει αν είναι [άρτιος ή περιττός](#).



Άσκηση 19^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει το Επίθετο ενός μαθητή, τις [απουσίες](#) του στα 2 τετράμηνα και στη συνέχεια θα υπολογίζει και θα εμφανίζει το Σύνολο των απουσιών. Τέλος, θα ελέγχει και θα εμφανίζει μήνυμα για το αν ο μαθητής μπορεί να συμμετέχει ή όχι στις εξετάσεις.



Στη Python, η συνάρτηση **round** στρογγυλοποιεί έναν αριθμό στον πλησιέστερο ακέραιο. Εάν ο αριθμός βρίσκεται ακριβώς στα μισά μεταξύ δύο ακεραίων αριθμών, στρογγυλοποιείται στον πλησιέστερο ζυγό (άρτιο) ακέραιο. Μελέτησε τις εκφράσεις.

```
>>> round(9.4)
9
>>> round(9.7)
10
>>> round(9.5)
10
>>> round(8.5)
8
```



Άσκηση 20^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάσει το Επίθετο ενός μαθητή, τη βαθμολογία του στα 2 τετράμηνα και στις γραπτές εξετάσεις και στη συνέχεια θα υπολογίζει και θα εμφανίζει τον [Μέσο Όρο της βαθμολογίας](#). Τέλος, θα ελέγχει και θα εμφανίζει μήνυμα για το αν ο μαθητής προάγεται ή παραπέμπεται.



Άσκηση 21^η Σε έναν διαγωνισμό στην Ελληνική Αστυνομία οι άντρες υποψήφιοι για να πετύχουν θα πρέπει να συγκεντρώσουν περισσότερους από 70 βαθμούς και να έχουν ύψος μεγαλύτερο από 175cm. Οι γυναίκες υποψήφιες θα πρέπει να συγκεντρώσουν περισσότερους από 60 βαθμούς και να έχουν ύψος μεγαλύτερο από 165cm. Να «γραφεί» πρόγραμμα που θα ελέγχει αν ένας/μία υποψήφιος/α πέτυχε ή όχι στον [διαγωνισμό](#).



Άσκηση 22^η Να «γραφεί» πρόγραμμα το οποίο θα υπολογίζει τη [συνάρτηση](#) $f(x) = \frac{x^2+1}{x}$



Άσκηση 23^η Να «γραφεί» πρόγραμμα το οποίο θα διαβάζει τη μεγαλύτερη πλευρά ενός τριγώνου (α) και τις 2 μικρότερες (β και γ) και στη συνέχεια θα υπολογίζει και θα εμφανίζει μήνυμα για το αν το τρίγωνο είναι ορθογώνιο η όχι ([αντίστροφο Πυθαγορείου θεωρήματος](#)).

Πολλαπλή επιλογή



Στην δομή πολλαπλής επιλογής έχουμε πολλές συνθήκες οι οποίες εξετάζονται μια προς μια μέχρι κάποια να ισχύει (True).

if <συνθήκη1>:

 σύνολο_εντολών1

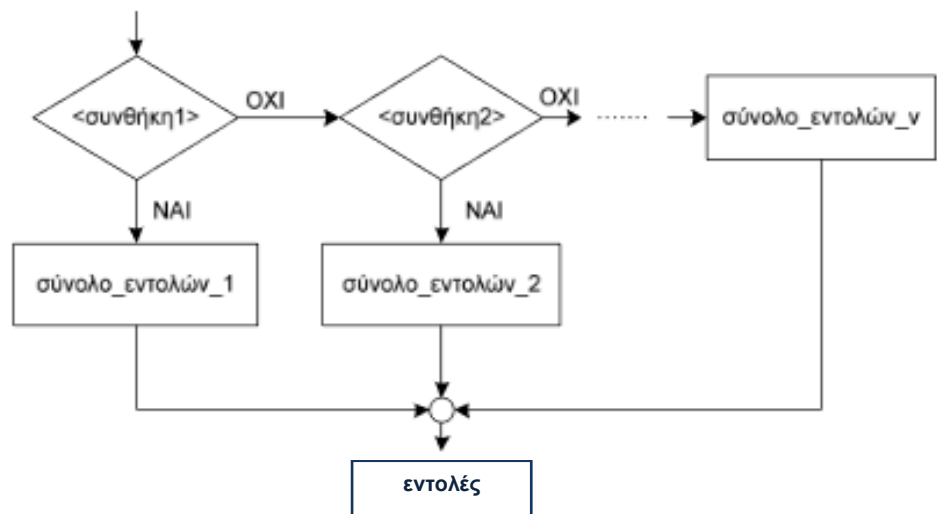
elif <συνθήκη2>:

 μπλοκ_εντολών2

else:

 μπλοκ_εντολών3

εντολές



το *elif* είναι μια συντομογραφία του *else if*

Μελέτησε το παράδειγμα:

```
temperature=float(input('Δώσε τη θερμοκρασία περιβάλλοντος'))
if temperature<8:
    print('Έχει παγωνιά')
elif temperature<15:
    print('Έχει κρύο')
elif temperature<30:
    print('Έχει καλό καιρό')
else:
    print('Έχει πολλή ζέστη')
```

Εμφωλευμένη επιλογή



Την εμφωλευμένη επιλογή τη χρησιμοποιούμε όταν θέλουμε να ελέγξουμε μια κατάσταση μέσα σε μια άλλη κατάσταση. Η χρήση εμφωλευμένων δομών απαιτεί ιδιαίτερη προσοχή, καθώς αυξάνεται η πολυπλοκότητα του αλγορίθμου.



Άσκηση 24^η Μελέτησε την επίλυση εξίσωσης 1^{ου} βαθμού ($ax+b=0$) με απλή, με πολλαπλή και με εμφωλευμένη επιλογή.

Με [απλή επιλογή](#)

```
# Επίλυση εξίσωσης 1ου βαθμού  $ax+b=0$ 
a=int(input('Δώσε το α:'))
b=int(input('Δώσε το β:'))
if a!= 0:
    x = -b / a
    print('x=',x)
if a==0 and b == 0:
    print('Η εξίσωση είναι αόριστη (ταυτότητα)')
if a==0 and b != 0:
    print('Η εξίσωση είναι αδύνατη')
```

Με [πολλαπλή επιλογή](#)

```
# Επίλυση εξίσωσης 1ου βαθμού  $ax+b=0$ 
a=int(input('Δώσε το α:'))
b=int(input('Δώσε το β:'))
if a!= 0:
    x = -b / a
    print('x=',x)
elif b == 0:
    print('Η εξίσωση είναι αόριστη (ταυτότητα)')
else:
    print('Η εξίσωση είναι αδύνατη')
```

Με [εμφωλευμένη επιλογή](#)

```
#Εξίσωση 1ου βαθμού  $ax+b=0$ 
a=int(input('Δώσε το α:'))
b=int(input('Δώσε το β:'))
if a!= 0:
    x = -b / a
    print('x=',x)
else:
    if b == 0:
        print('Η εξίσωση είναι αόριστη (ταυτότητα)')
    else:
        print('Η εξίσωση είναι αδύνατη')
```

Ε.3 Δομή Επανάληψης ή Επαναληπτική Δομή



Η **Δομή Επανάληψης** είναι ιδιαίτερα συχνή και εφαρμόζεται σε ένα πλήθος προβλημάτων στα οποία θα πρέπει να επαναληφθούν κάποιες εντολές, πολλές φορές.

Η εντολή **for**



Χρησιμοποιούμε την εντολή **for** όταν οι επαναλήψεις τερματίζουν μετά από ένα προκαθορισμένο αριθμό. Π.χ. για τον σχεδιασμό ενός τετραγώνου χρειάζονται 4 επαναλήψεις, για το πεντάγωνο 5 επαναλήψεις κ.ο.κ

```
from turtle import
kate = Turtle()
kate.pendown()
for i in range(4):
    kate.forward(100)
    kate.left(90)
```

Μελέτησε τις παρακάτω εντολές και τα αποτελέσματα:

<pre>for i in range(5): print(i)</pre>	0 1 2 3 4
<pre>for i in range(1,7): print(i)</pre>	1 2 3 4 5 6
<pre>for i in range(1,10, 2): print(i)</pre>	1 3 5 7 9
<pre>for i in range(10,0,-2): print(i)</pre>	10 8 6 4 2



Η συνάρτηση **range()** δημιουργεί ακολουθίες αριθμών και έχει την ακόλουθη σύνταξη:

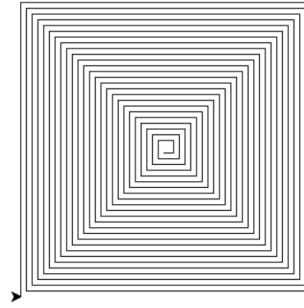
range(αρχή, τέλος, βήμα) όπου αρχή, τέλος και βήμα είναι ακέραιοι αριθμοί. Μπορούμε να καλέσουμε την συνάρτηση range με 3 διαφορετικούς τρόπους:

1. **range(τέλος)** Δίνει μια ακολουθία αριθμών από το 0 έως και την τιμή «τέλος-1»
2. **range(αρχή, τέλος)** Δίνει μια ακολουθία αριθμών από την τιμή «αρχή» μέχρι την τιμή «τέλος-1»
3. **range(αρχή, τέλος, βήμα)** Δίνει μια ακολουθία αριθμών από την τιμή «αρχή» μέχρι την τιμή «τέλος-1» και με βήμα την τιμή «βήμα»



Άσκηση 25^η Μελέτησε τον κώδικα που εμφανίζει το [σπирάλ](#) της εικόνας.

```
from turtle import *
franklin = Turtle()
franklin.goto(0, 0)
franklin.pendown()
length = 10
for i in range(100):
    franklin.forward(length)
    franklin.left(90)
    length += 3
```



Άσκηση 26^η Η ακολουθία **Fibonacci** είναι μια ακολουθία αριθμών που εμφανίζεται στη φύση, όπως για παράδειγμα στη διάταξη των φύλλων ενός φυτού και στα πέταλα των λουλουδιών. Οι σπείρες των κοχυλιών και των σαλιγκαριών συχνά ακολουθούν την ακολουθία Fibonacci. Το αρχικό παράδειγμα που χρησιμοποίησε ο Fibonacci για να περιγράψει την ακολουθία του είναι η αναπαραγωγή κουνελιών. Κάθε ζεύγος κουνελιών παράγει ένα νέο ζεύγος κάθε μήνα, με κάθε νέο ζεύγος να ξεκινά την αναπαραγωγή από τον δεύτερο μήνα. Έτσι έχουμε τους αριθμούς 1, 1, 2, 3, 5, 8, 13, 21 κ.λπ. Κάθε αριθμός στην ακολουθία είναι το άθροισμα των δυο προηγούμενων.

Να «γραφεί» το πρόγραμμα το οποίο θα εμφανίζει την παρακάτω ακολουθία [Fibonacci](#) με χρήση της εντολής **for**.

1 1 2 3 5 8 13 21 34 55 89 144



Άσκηση 27^η Το παραγοντικό ενός φυσικού αριθμού n , συμβολίζεται με $n!$ και είναι το γινόμενο όλων των θετικών ακεραίων μικρότερων ή ίσων με n .

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$$

Για παράδειγμα:

$$2! = 1 \cdot 2 = 2$$

$$3! = 1 \cdot 2 \cdot 3 = 6$$

$$4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$$

Μελέτησε τον κώδικα για τον παραγοντικό [factorial n!](#). Τρέξε το πρόγραμμα για διάφορες τιμές του n .

Η εντολή while



Χρησιμοποιούμε την εντολή **while** όταν οι επαναλήψεις τερματίζουν μετά από έλεγχο μιας συνθήκης.



Άσκηση 28^η Ο/Η ταμίας σε ένα super market καταχωρεί στην ταμειακή απόδειξη την τιμή του κάθε προϊόντος. Η διαδικασία ολοκληρώνεται μόλις εισαχθεί η τιμή 0. Να «γραφεί» το πρόγραμμα το οποίο θα εκτυπώνει το συνολικό ποσό της αγοράς.



Η συνάρτηση **randint(αρχική_τιμή, τελική_τιμή)** επιστρέφει έναν τυχαίο ακέραιο από μια αρχική τιμή έως μια τελική τιμή. Για να χρησιμοποιήσουμε τη **randint** θα πρέπει πρώτα να εισαχθεί η βιβλιοθήκη **random**.

```
import random
a=random.randint(3,6) #θα επιστρέψει έναν τυχαίο αριθμό από το 3 έως το 6
print(a)
```



Άσκηση 29^η Να «γραφεί» πρόγραμμα το οποίο θα υλοποιεί το παρακάτω παιχνίδι τύχης. Ο υπολογιστής «επιλέγει» τυχαία έναν αριθμό (από το 1 μέχρι το 10) και ένας παίκτης προσπαθεί να μαντέψει τον αριθμό αυτό. Μόλις ο παίκτης βρει τον αριθμό εμφανίζεται μήνυμα για το πόσες προσπάθειες χρειάστηκαν.



Άσκηση 30^η Να «γραφεί» πρόγραμμα το οποίο θα προσομοιώνει το παιχνίδι πέτρα, ψαλίδι, χαρτί.

- για 1 παιχνίδι
- για 4 παιχνίδια

ΣΤ. Συναρτήσεις



Μια συνάρτηση μας επιτρέπει να δίνουμε ένα όνομα σε μια ομάδα εντολών και να την εκτελούμε χρησιμοποιώντας το όνομά της οπουδήποτε στο πρόγραμμά μας και για όσες φορές θέλουμε. Επίσης, μια συνάρτηση μπορεί να δεχθεί είσοδο και να παραγάγει έξοδο. Ο ορισμός μιας νέας συνάρτησης μπορεί να κάνει ένα πρόγραμμα μικρότερο, απομακρύνοντας τμήματα κώδικα που επαναλαμβάνονται. Στην Python οι συναρτήσεις ορίζονται με τη δεσμευμένη λέξη **def**.

def name():, όπου name το όνομα της συνάρτησης.



Άσκηση 31^η Να «γραφεί» πρόγραμμα το οποίο θα σχεδιάζει 4 πολύγωνα με πλευρά 50 (1 πεντάγωνο, 1 εξάγωνο, 1 επτάγωνο και 1 οκτάγωνο) με χρήση συνάρτησης



Άσκηση 32^η Να «γραφεί» πρόγραμμα το οποίο θα σχεδιάζει 12 πολύγωνα, ξεκινώντας από ένα τετράγωνο και καταλήγοντας σε ένα δεκαπεντάγωνο, αυξάνοντας το πλήθος των πλευρών κατά 1 κάθε φορά, γράφοντας την εντολή Polygon μόνο μια φορά.