

Εισαγωγή στη
Γλώσσα Προγραμματισμού



Γ' Γυμνασίου
Σημειώσεις και Ασκήσεις του
Γιώργου Ρούσσα

Τμήμα: _____

Σταθμός Εργασίας: _____

Ονοματεπώνυμο Μαθητών:

Λίγα λόγια για τη γλώσσα προγραμματισμού Python

Η γλώσσα προγραμματισμού Python είναι από τις σημαντικότερες και πιο διαδεδομένες γλώσσες προγραμματισμού σήμερα κυρίως λόγω της ευρείας γκάμας εφαρμογών της, της ευκολίας εκμάθησής της, της παιδαγωγικής της αξίας και των δυνατοτήτων που έχει στη διαχείριση δεδομένων πολλών τύπων.

Στα σημαντικότερα πλεονεκτήματά της συμπεριλαμβάνονται:

- η δωρεάν διάθεσή της και η εύκολη εγκατάστασή της στον υπολογιστή μας
- μεγάλος αριθμός online Περιβαλλόντων Προγραμματισμού που την υποστηρίζουν (edublocks, online python, spyder, Visual Studio Code, codewith, **Thonny**)
- η χρήση Διερμηνευτή-interpreter (για μεμονωμένη εκτέλεση εντολών στο Κέλυφος) και Μεταγλωττιστή-compiler (για εκτέλεση ολόκληρου προγράμματος) στο ίδιο προγραμματιστικό περιβάλλον
- ο τεράστιος αριθμός εκπαιδευτικών videos που είναι διαθέσιμα στο διαδίκτυο και οι πολλές ελεύθερες πλατφόρμες εκπαίδευσης που διατίθενται δωρεάν.

Η Python, δημιουργήθηκε από τον Ολλανδό Γκίντο βαν Ρόσσουμ (Guido van Rossum) στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) το 1989 και κυκλοφόρησε για πρώτη φορά το 1991. Αναπτύσσεται ως ανοιχτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL.

Το όνομα της γλώσσας προέρχεται από την ομάδα των Άγγλων κωμικών Μόντυ Πάιθον (Monty Python) και δεν έχει καμιά σχέση με το φίδι πύθωνα, παρότι το λογότυπό της παραπέμπει σε κάτι τέτοιο.

Τύποι Δεδομένων

Τα δεδομένα που αποθηκεύονται στη μνήμη για να τα επεξεργαστούμε μπορεί να είναι διαφορετικών τύπων, που πρακτικά σημαίνει, ότι απαιτούνται λιγότερες ή περισσότερες θέσεις μνήμης αλλά και τρόπο που μπορούμε να τα επεξεργαστούμε και επενεργήσουμε σε αυτά.

Τύπος	Κλάση	Παραδείγματα
Ακέραιος	int	1, 0, -1, 1024
Κινητής Υποδιαστολής	float	3.14159, 0.5, 3.0
Λογικός	bool	True, False
Αλφαριθμητικός	str	"Πυργετός", "Κάρτσωνα 1"
Λίστα	list	[1, 1, 2, 3, 5, 8], ["a", "34w", 45, True]
Πλειάδα	tuple	("Πυργετός", "Αιγάνη", "Ομόλιο", "Στόμιο")
Απουσία τιμής	none	None

- Αριθμητικοί τύποι είναι οι **ακέραιοι** (int-integer) και οι **κινητής υποδιαστολής** (float) – η υποδιαστολή είναι με τελεία ('.') και όχι με κόμμα.
- Οι **λογικοί** τύποι (bool-boolean) δέχονται μόνο δύο τιμές True ή False, το αποτέλεσμα ενός ελέγχου.
- Οι **συμβολοσειρές** (str-string) ή αλφαριθμητικά είναι μια ακολουθία από γράμματα, αριθμούς, σύμβολα κλεισμένα μέσα σε μονά (' ') ή διπλά (" ") εισαγωγικά.
- Οι **λίστες** (list) είναι ο πιο σημαντικός τύπος και είναι σύνθετος, δηλαδή μπορεί να περιέχει ταυτόχρονα πολλά και διαφορετικά αντικείμενα οποιουδήποτε τύπου.

Μπορούμε να ελέγξουμε τον τύπο δεδομένων μιας έκφρασης με την εντολή **type()** όπως φαίνεται πιο κάτω χρησιμοποιώντας το Κέλυφος του **Thonny**.



```
>>> type(8192)
<class 'int'>
>>> type(-5.2119)
<class 'float'>
>>> type(True)
<class 'bool'>
>>> type("Πυργετός")
<class 'str'>
>>> type([8192, True, "Αιγάνη"])
<class 'list'>
>>> type(("Πυργετός", 40007))
<class 'tuple'>
```

Δραστηριότητα

Συμπληρώστε στον πίνακα που ακολουθεί τον κατάλληλο τύπο δεδομένων και κλάση του. (Χρησιμοποιήστε αν θέλετε το Κέλυφος του **Thonny**)



Τιμή	Τύπος	Κλάση
-5.1		
2		
"Nik.G."		
0.0		
"3.2"		
[3,2]		

Τιμή	Τύπος	Κλάση
["Γυμνάσιο", "Μαθητές", 85]		
"True"		
False		
[False, "True", None]		
5+3/2		
5+3%2		

Αριθμητικές εκφράσεις

Οι αριθμητικές εκφράσεις χρησιμοποιούν αριθμητικούς τελεστές και αριθμητικές μεταβλητές ή σταθερές τιμές αριθμητικού τύπου. Οι εκφράσεις αυτές υπολογίζονται σύμφωνα με την προτεραιότητα των πράξεων που φαίνεται στον παρακάτω πίνακα και όπου απαιτείται αλλαγή αυτής της προτεραιότητας χρησιμοποιούμε ζευγάρια παρενθέσεων.

Προτεραιότητα αριθμητικών πράξεων	
()	Αλλαγή της προτεραιότητας (οι πράξεις μέσα στις παρενθέσεις προηγούνται όλων των υπόλοιπων πράξεων)
**	Ύψωση σε δύναμη
* / // %	Πολλαπλασιασμός, Διαίρεση, Ακέραια Διαίρεση, Υπόλοιπο Ακέραιας Διαίρεσης
+ -	Πρόσθεση, Αφαίρεση
Παρατήρηση: Όταν στην έκφραση υπάρχουν δύο ή περισσότερες αριθμητικές πράξεις της ίδιας προτεραιότητας αυτές υπολογίζονται από αριστερά προς τα δεξιά όπως εμφανίζονται	

Μελετήστε τα παραδείγματα πιο κάτω και τα αποτελέσματα που προκύπτουν.

```
>>> 5+3-6
2
>>> 3**2 + 1
10
>>> 32/4
8.0
>>> 32//4
8
```

```
>>> 4**(1/2)
2.0
>>> 4**2/4
4.0
>>> 49%6
1
```



Παρατήρηση: Αν θέλω να καθαρίσω το Κέλυφος πατάω συγχρόνως Ctrl+L

Δραστηριότητα

Συμπληρώστε στον πίνακα που ακολουθεί το αποτέλεσμα της έκφρασης και τον αντίστοιχο αριθμητικό τύπο του. (Χρησιμοποιήστε αν θέλετε το Κέλυφος του Thonny για να επιβεβαιώσετε το αποτέλεσμα που βρήκατε)



Έκφραση	Αποτέλεσμα	Αριθμητικός Τύπος
>>> 5+2*3-9		
>>> 3**3/10		
>>> 25//8		
>>> 18/4		
>>> 41%5		
>>> 31//5%4		
>>> 16-(3**2)/2		
>>> (3+5)*(5-3.0)		
>>> 2**(12//4)		
>>> 45%8//2		
>>> -1**2		

Παρατήρηση: Σχηματικά, η Ακέραια Διαίρεση και το Υπόλοιπο της Ακέραιας Διαίρεσης, φαίνεται στο παρακάτω σχήμα.

$$\begin{array}{r|l}
 29 & 6 \\
 \hline
 5 & 4
 \end{array}$$

$29\%6$ $\longrightarrow$ 5 $\longleftarrow$ $29//6$

Σχεσιακές και Λογικές εκφράσεις

Οι σχεσιακές και λογικές εκφράσεις αποδίδουν πάντα ένα λογικό αποτέλεσμα, δηλαδή ένα αποτέλεσμα που μπορεί να είναι ή αληθές (True) ή ψευδές (False) και χρησιμοποιούνται κυρίως στη λήψη αποφάσεων.

Οι **σχεσιακές εκφράσεις** χρησιμοποιούν τους τελεστές της σύγκρισης ανάμεσα σε αριθμούς, αλφαριθμητικά και λογικά τελούμενα (σταθερές αριθμητικές ή αλφαριθμητικές τιμές, μεταβλητές ή το αποτέλεσμα εκφράσεων-αριθμητικών ή άλλων).

Σχεσιακοί τελεστές				
<	Μικρότερο από		>=	Μεγαλύτερο ή ίσο από
<=	Μικρότερο ή ίσο από		==	Ίσο με
>	Μεγαλύτερο από		!=	Διάφορο από
Παρατήρηση: Το αποτέλεσμα που παράγεται από μία σύγκριση μπορεί να είναι ή αληθές (True) ή ψευδές (False)				

Αν οι συγκρίσεις γίνονται ανάμεσα σε αλφαριθμητικά, τότε ακολουθείται η αλφαβητική σειρά που γνωρίζουμε ανά σύμβολο με την κατάταξή τους ως προς τον πίνακα ASCII και ως προς την αντίστοιχη θέση.

Οι **λογικές εκφράσεις** χρησιμοποιούν τους λογικούς τελεστές ανάμεσα σε λογικές τιμές (True ή False) ή μεταβλητές λογικού τύπου ή εκφράσεις που το αποτέλεσμά τους είναι λογικού τύπου. Υπάρχουν τρεις λογικοί τελεστές.

Λογικοί τελεστές	
not (όχι)	Λογική πράξη άρνησης
and (και)	Λογική πράξη σύζευξης
or (ή)	Λογική πράξη διάζευξης
Παρατήρηση: Το αποτέλεσμα που παράγεται από μία λογική πράξη είναι ή αληθές (True) ή ψευδές (False)	

Για να κατανοήσουμε το αποτέλεσμα κάθε μιας από τις παραπάνω λογικές πράξεις δίνεται ο παρακάτω πίνακας με τα αποτελέσματα κάθε μιας.

A	B	A and B	A or B	not A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Παρατήρηση: Για την ευκολότερη απομνημόνευση του παραπάνω πίνακα θυμόμαστε ότι «στη λογική **διάζευξη** (or) υπερισχύει η Αλήθεια (True), ενώ στη λογική **σύζευξη** (and) υπερισχύει το Ψεύδος (False)»

Μελετήστε τα παραδείγματα πιο κάτω και τα αποτελέσματα που προκύπτουν.



```
>>> 763==763
True
>>> 5!=9
True
>>> 763<=819
True
>>> 763<=2
False
>>> (16<13) and (5!=8)
False
>>> (16<13) or (5!=8)
True
>>> not (763<=2)
True
```


Δραστηριότητα

Συμπληρώστε στον πίνακα που ακολουθεί το αποτέλεσμα της έκφρασης. (Χρησιμοποιήστε το Κέλυφος του Thonny για να επιβεβαιώσετε το αποτέλεσμα που βρήκατε)



Έκφραση	Αποτέλεσμα
>>> 3<=5	
>>> 5+4!=3**2	
>>> 3!=(5+1) and True	
>>> '51'<'5'	
>>> False==False	
>>> True and 3>=9	
>>> 'Αθηνά'<='Πέτρος'	
>>> 'Άννα'!='Άνα'	
>>> (3<5) and 'N'<'Nik'	
>>> False>True	

Αλφαριθμητικές εκφράσεις

Οι αλφαριθμητικές εκφράσεις χρησιμοποιούν τον τελεστή '+' για να συνενώνει (προσθέτει) δύο ή περισσότερα αλφαριθμητικά, καθώς και τον τελεστή '*' για να επαναλαμβάνει ένα αλφαριθμητικό περισσότερες φορές.

Αν θέλουμε να μετατρέψουμε σε αλφαριθμητικό κάποια άλλου τύπου τιμή (αριθμική ή λίστα) μπορούμε να χρησιμοποιήσουμε τη συνάρτηση str() που μας επιστρέφει ό,τι δώσαμε σαν παράμετρο σε μορφή αλφαριθμητικού μέσα σε ' ' ή " ".

Μελετήστε τα παραδείγματα πιο κάτω και τα αποτελέσματα που προκύπτουν.



```
>>> 'Inter' + 'national'
'International'
>>> 'Δήμοι ' + 'Τεμπών και' + ' Αγιάς'
'Δήμοι Τεμπών και Αγιάς'
>>> 3 * 'Σιγά '
'Σιγά Σιγά Σιγά '
>>> (2**3) * 'Λα'
'ΛαΛαΛαΛαΛαΛαΛα'
>>> 2 * 'Όχι' +2
TypeError
>>> 'Κούμα ' + str(11)
'Κούμα 11'
>>> (2 * 'Α'+str(4) ) * 3
'AA4AA4AA4'
```

Δραστηριότητα

Συμπληρώστε στον πίνακα που ακολουθεί το αποτέλεσμα της έκφρασης. (Χρησιμοποιήστε το Κέλυφος του Thonny για να επιβεβαιώσετε το αποτέλεσμα που βρήκατε)



Έκφραση	Αποτέλεσμα
>>> 'A'+'B'	
>>> 'str(3)'	
>>> str(2)+'025'	
>>> 2*str(9)+' red balloons'	
>>> 'Android '+str(12.5)	
>>> str(2*3)+'17'+str(4)	
>>> 3*'Angel'[2]	

Μεταβλητές

Για να αποθηκεύσουμε προσωρινά τα δεδομένα μας και να έχουμε πρόσβαση σε αυτά χρησιμοποιούμε μεταβλητές. Κάθε μεταβλητή αντιστοιχεί σε μια θέση μνήμης στον υπολογιστή μας και μπορεί να πάρει τιμές από διαφορετικούς τύπους δεδομένων. Στη μνήμη τα δεδομένα μας αναπαριστούνται με δυαδικά ψηφία 0 και 1 και επομένως μια ακολουθία δυαδικών ψηφίων μπορεί να ερμηνεύεται με διαφορετικό τρόπο ανάλογα με τον τύπο των δεδομένων που την «βλέπει». Κάθε μεταβλητή έχει πρόσβαση στα δεδομένα της μνήμης στα οποία αναφέρεται και έχει δεσμευτεί γι' αυτή.

Για τη χρήση μνήμης από μια μεταβλητή δεν είναι απαραίτητη η δήλωσή της ούτε και η δήλωση του τύπου της. Κάθε φορά που θέλουμε να συσχετίσουμε μια μεταβλητή με δεδομένα άλλου τύπου μπορούμε να το πράξουμε χωρίς περιορισμούς.

Μελετήστε τα παραδείγματα πιο κάτω και τα αποτελέσματα που προκύπτουν.

```
>>> a=6
>>> print(a,b,c)
6 8 18
```



```
>>> a=5
>>> b=8
>>> a+b
13

>>> c=2*a+b
>>> print(a,b,c)
5 8 18

>>> a=6
>>> c=a
>>> print(a,b,c)
6 8 6
```

Δραστηριότητα

Συμπληρώστε στο κέλυφος που ακολουθεί τα αποτελέσματα της εκτύπωσης όπου υπάρχει υπογράμμιση δίπλα από την εντολή print(). Στην εκτύπωση περισσότερων από μία τιμών υπάρχει ένα κενό ανάμεσα στις τιμές των παραμέτρων. (Χρησιμοποιήστε το Κέλυφος του Thonny για να επιβεβαιώσετε τα αποτελέσματα που βρήκατε)



```
>>> a=b=3
>>> c=b=4
>>> print(a,b,c) _____

>>> b=d=c-a
>>> print(a,b,c,d) _____

>>> e=b
>>> b=c
>>> c=e
>>> print(a,b,c,d) _____
```

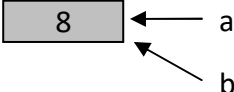
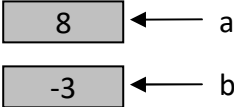
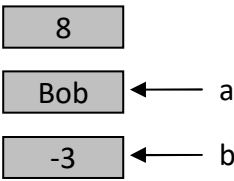
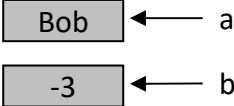
Δραστηριότητα

Συμπληρώστε στο κέλυφος που ακολουθεί τα αποτελέσματα της εκτύπωσης όπου υπάρχει υπογράμμιση δίπλα από την εντολή print(). (Χρησιμοποιήστε το Κέλυφος του Thonny για να επιβεβαιώσετε τα αποτελέσματα που βρήκατε)



```
>>> a=2
>>> b='Leo'
>>> c=3/a
>>> print(a,b,c) _____
>>> c=a*b
>>> print(-a,b,c) _____
>>> b=c
>>> c=str(a*a)+b
>>> print(a,b,c) _____
```

Παράδειγμα συσχετίσεων μεταβλητών με δεδομένα

Ενέργεια	Εντολή	Αποτύπωση μνήμης
Η μεταβλητή a παίρνει την τιμή 8	a=8	
Η μεταβλητή b παίρνει την τιμή της μεταβλητής a	b=a	
Η μεταβλητή b παίρνει την τιμή -3	b=-3	
Η μεταβλητή a παίρνει την τιμή 'Bob'	a='Bob'	
Η Python διαγράφει αυτόματα όλα τα δεδομένα (τιμές) που δεν δεικτοδοτούνται από κάποια μεταβλητή		

Δραστηριότητα

Σχεδιάστε την αποτύπωση της μνήμης για την παρακάτω ακολουθία εντολών. (Χρησιμοποιήστε το περιβάλλον του Thonny και αφού αποθηκεύσετε προσωρινά το πρόγραμμα πατάτε Ctrl+F5 και επαναλαμβανόμενα το F7 για να το παρακολουθήσετε στην εξέλιξή του κοιτάζοντας και το παράθυρο Μεταβλητές)



Εντολή	Αποτύπωση μνήμης
<pre>x=y=z=1 y='Python' z=x+1 x=d=y</pre>	

Τα ονόματα των μεταβλητών ακολουθούν κάποιους κανόνες σύνταξης:

- ξεκινούν πάντα με γράμμα ή κάτω παύλα (_ ή underscore)
- υπάρχει διάκριση ανάμεσα σε κεφαλαία και πεζά
- επιτρέπονται ελληνικοί χαρακτήρες (δεν είναι πάντα επιθυμητό)
- επιτρέπονται εκτός από γράμματα, αριθμοί και κάτω παύλα, αλλά απαγορεύεται ο κενός χαρακτήρας (space) ανάμεσα στους υπόλοιπους χαρακτήρες
- δεν επιτρέπονται δεσμευμένες λέξεις της Python.

Μελετήστε τα παραδείγματα πιο κάτω και τα αποτελέσματα που προκύπτουν για τα ονόματα των μεταβλητών



```
>>> a=5          # Ένα γράμμα για όνομα μεταβλητής
>>> _b=6         # Κάτω παύλα και γράμμα για όνομα μεταβλητής
>>> a-b=8        # Δεν επιτρέπεται το σύμβολο – μέσα στο όνομα της μεταβλητής
File "<stdin>", line 1
  a-b=8
  ^^
SyntaxError: cannot assign to expression here. Maybe you meant '==' instead of '='?

>>> a13b=14      # Γράμματα και αριθμοί στο όνομα μιας μεταβλητής
>>> gia ola=6    # Δεν επιτρέπεται το κενό μέσα στο όνομα της μεταβλητής
File "<stdin>", line 1
  gia ola=6
  ^^
SyntaxError: invalid syntax

>>> gia_liga=4   # Μόνο σύμβολο που επιτρέπεται είναι το underscore _
>>>
```

Παρατήρηση: Αν θέλουμε να εισάγουμε ένα σχόλιο για να περιγράψουμε κάποια λειτουργία ή ενέργειά μας μπορούμε να χρησιμοποιήσουμε το σύμβολο # και το κείμενο που βρίσκεται δεξιά του δεν εκτελείται από τον διερμηνευτή της Python

Δραστηριότητα

Συμπληρώστε στον πίνακα που ακολουθεί ποιες μεταβλητές είναι αποδεκτές και ποιες όχι ως προς το όνομά τους. Συμπληρώστε δίπλα τον λόγο που δεν είναι αποδεκτή.



Όνομα Μεταβλητής	Αποδεκτό Μη αποδεκτό	Μη αποδεκτό γιατί...
George		
Nik Galis		
__a_b		

Pan@gmail		
Αργεντίνια		
2025_Val		
Year2025_		

Εντολή εξόδου print()

Για να εμφανίσουμε το περιεχόμενο μιας ή περισσότερων μεταβλητών καθώς και άλλες πληροφορίες μπορούμε να χρησιμοποιήσουμε την εντολή print() και μέσα στην παρένθεση να τοποθετήσουμε τα περιεχόμενα που θέλουμε να εμφανιστούν.

Εξ ορισμού, εφόσον υπάρχουν περισσότερα από ένα πράγματα να εμφανιστούν, αυτά χωρίζονται με έναν κενό χαρακτήρα (space) ενώ στο τέλος αυτόματα αλλάζει γραμμή (\n). Μπορούμε να αλλάξουμε τη γραμμή που εκτυπώνουμε κάνοντας χρήση του ειδικού χαρακτήρα new line (\n) όταν θελήσουμε ή να την απενεργοποιήσουμε.

Μελετήστε τα παραδείγματα πιο κάτω και τα αποτελέσματα που προκύπτουν.



```
>>> a=3
>>> b=6
>>> p='Python'
>>> print(a,b,p)                # Έξοδος 1
3 6 Python

>>> print(a,b,p,sep='-')        # Έξοδος 2
3-6-Python

>>> print(a,b,p,sep='')         # Έξοδος 3
36Python

>>> print(a,b,p);print(a,b)     # Έξοδος 4
3 6 Python
3 6
```



```
>>> print(a,b,p,end="");print(a,b)          # Έξοδος 5
3 6 Python3 6

>>> print(a,b,p,end='new line\n');print(a,b) # Έξοδος 6
3 6 Pythonnew line
3 6
```

Παρατήρηση: Αν θέλουμε να εκτελέσουμε περισσότερες από μία εντολές γράφοντάς τις τη μία δίπλα στην άλλη, πρέπει να τοποθετήσουμε ανάμεσα στις εντολές τον διαχωριστή (;)

Δραστηριότητα

Συμπληρώστε τί θα εκτυπωθεί στο κέλυφος που ακολουθεί μετά την εκτέλεση των εντολών print() με τις παραμέτρους που έχουν. (Χρησιμοποιήστε το Κέλυφος του Thonny για να επιβεβαιώσετε τα αποτελέσματα που βρήκατε)



```
>>> x='Year'
>>> c=20; y=25
>>> print(x,c,y)

>>> print(x,c,y,sep=")          # sep είναι ο διαχωριστής ανάμεσα στις-παραμέτρους

>>> print(x,' ',c,y,sep=")

>>> print(x,'\n',c,y,sep=")      # \n είναι ο ειδικός χαρακτήρας αλλαγής γραμμής

>>> print(x); print(c,y,sep=")
```

```
print(x,end=' '); print(c,y,sep=")    # end είναι ο χαρακτήρας στο τέλος της εκτύπωσης
```

Η εντολή $x=x+1$

Πολύ συχνά απαιτείται κατά τον προγραμματισμό να αυξήσουμε ή να μειώσουμε την τιμή μιας μεταβλητής. Η εντολή $x=x+1$, που εκ πρώτης όψης δεν φαίνεται σωστή από μαθηματικής σκοπιάς, για την Python, αλλά και τις περισσότερες γλώσσες προγραμματισμού είναι αυτή που αυξάνει την τιμή της μεταβλητής x κατά 1, ή αν η σταθερή τιμή είναι άλλη, κατά την ποσότητα αυτής της τιμής.

Πρακτικά, αφού υπολογιστεί η τιμή της έκφρασης στα δεξιά, αντικαθιστώντας τις μεταβλητές που εμφανίζονται με τις αντίστοιχες σταθερές τους ποσότητες, το αποτέλεσμα μεταβιβάζεται στην μεταβλητή που βρίσκεται στο αριστερό μέρος και αντικαθιστά την μέχρι εκείνη τη στιγμή ποσότητά της.

Μελετήστε τα παραδείγματα πιο κάτω και τα αποτελέσματα που προκύπτουν.



```
>>> x=5
>>> x=x+1      # Η τιμή της μεταβλητής x αυξάνεται κατά 1
>>> print (x)
6
>>> x=x-2      # Η τιμή της μεταβλητής x μειώνεται κατά 2
>>> print (x)
4
>>> x+=2       # Εναλλακτικός τρόπος να αυξηθεί η x κατά 2
>>> print (x)
6
>>> x-=1       # Η τιμή της μεταβλητής x μειώνεται κατά 1
>>> print (x)
5
```

Δραστηριότητα

Εκτελέστε τις παρακάτω εντολές στο κέλυφος και βρείτε περιγράψτε τη λειτουργία που πραγματοποιείται και τί θα εκτυπωθεί μετά την εκτέλεση των εντολών `print()` με τις παραμέτρους που έχουν. (Χρησιμοποιήστε το Κέλυφος του Thonny για να επιβεβαιώσετε τα αποτελέσματα που βρήκατε)



```
>>> a=61
>>> b=74
>>> a=a+b
>>> b=a-b
>>> a=a-b
>>> print (a,b)
```

```
>>> a=61
>>> b=74
>>> temp=a
>>> a=b
>>> b=temp
>>> print (a,b)
```

Είσοδος Δεδομένων με την εντολή `input()`

Για να εισάγουμε δεδομένα από το πληκτρολόγιο σε μια μεταβλητή, χρησιμοποιούμε την εντολή `input()` που διακόπτει την εκτέλεση του προγράμματος για να εισαχθούν από το πληκτρολόγιο τα δεδομένα και να αποδοθούν στην αντίστοιχη μεταβλητή και η εκτέλεση του προγράμματος συνεχίζεται.

Εντολή	Επεξήγηση Εντολής και παραμέτρων
<code>x=input("Γράψε μου το όνομά σου:")</code>	Εμφανίζει το μήνυμα των εισαγωγικών στην οθόνη και περιμένει να πληκτρολογήσουμε μια αλφαριθμητική τιμή (str) και την αποθηκεύει στη μεταβλητή x. Παρατήρηση: Πάντοτε η <code>input()</code> διαβάζει τιμές σε αλφαριθμητική μορφή, ποτέ αριθμούς!

<code>a=int(input("Η ηλικία σου;"))</code>	Αν θέλουμε να διαβάσουμε κάποιον αριθμό με την εντολή <code>input()</code> , θα πρέπει να γίνει ταυτόχρονα και η μετατροπή του στο σωστό τύπο δεδομένων.
--	--

Μελετήστε τα παραδείγματα πιο κάτω και τα αποτελέσματα που προκύπτουν.



```
>>> a=int(input('Δώσε έναν ακέραιο αριθμό:')); print(a)
Δώσε έναν ακέραιο αριθμό:7                                # Πληκτρολογώ τον αριθμό 7
7
>>> f=float(input('Δώσε έναν πραγματικό αριθμό:')); print(f)
Δώσε έναν πραγματικό αριθμό:3                              # Πληκτρολογώ τον αριθμό 3
3.0
>>> print(a*f)
21.0
>>> m=int(input('Δώσε έναν ακέραιο αριθμό:')); print(m)
Δώσε έναν ακέραιο αριθμό:αριθμός                          # Πληκτρολογώ τη λέξη αριθμός
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'αριθμός' # Λάθος στη μετατροπή
```

Δραστηριότητα (prog-a01.py)



Ας γράψουμε το πρώτο μας πλήρες πρόγραμμα. Χρησιμοποιούμε το περιβάλλον του Thonny για να γράψουμε το πρόγραμμά μας και να το εκτελέσουμε. Ένα πλήρες πρόγραμμα το γράφουμε στο χώρο πάνω από το κέλυφος αλλά για να εκτελεστεί θα πρέπει προηγουμένως να το αποθηκεύσουμε, έστω και προσωρινά με ένα όνομα αρχείου που έχει επέκταση (.py – Χρησιμοποιήστε το όνομα **prog-a01.py**). Στη συνέχεια από την εργαλειοθήκη επιλέγουμε το πράσινο πλήκτρο του Play ή πατάμε F5.

- Να εισάγετε μια ακέραια τιμή στη μεταβλητή `x`
- Αυξήστε την τιμή της μεταβλητής κατά 1
- Διπλασιάστε την τιμή της μεταβλητής

- δ. Μειώστε την τιμή της μεταβλητής κατά 5
- ε. Αυξήστε την τιμή της κατά 60%
- στ. Τοποθετήστε την τετραγωνική ρίζα της μεταβλητής στη μεταβλητή tr και εκτυπώστε τη
- ζ. Υπολογίστε και τοποθετήστε στη μεταβλητή exgr το τετράγωνο του αθροίσματος της x και της tr και εκτυπώστε το

Μετά από κάθε βήμα να εμφανίζεται η τιμή της μεταβλητής x εφόσον μεταβάλλεται και οποιαδήποτε άλλη τιμή σας ζητείται προσθέτοντας και τα κατάλληλα διευκρινιστικά μηνύματα.

Ενσωματωμένες συναρτήσεις

Στην Python διατίθενται κάποιες ενσωματωμένες συναρτήσεις για τη μετατροπή δεδομένων από έναν τύπο σε άλλο (εφόσον κάτι τέτοιο είναι δυνατό), αλλά και κάποιες ακόμη που χειρίζονται τα δεδομένα μας με συγκεκριμένο τρόπο.

Ενσωματ. Συνάρτηση	Παράδειγμα	Περιγραφή
float()	>>>float(5) 5.0	Μετατρέπει μια τιμή σε δεκαδικό αριθμό
int()	>>>float(5.999) 5	Μετατρέπει οποιαδήποτε αριθμητική τιμή και τη μετατρέπει σε ακέραια με αποκοπή δεκαδικών ψηφίων
str()	>>>str(9)+str(999) '9999'	Μετατρέπει μια τιμή σε αφηριθμητικό
list()	>>>list('unicef') ['u', 'n', 'i', 'c', 'e', 'f']	Μετατρέπει ένα αντικείμενο σε λίστα εφόσον κάτι τέτοιο είναι δυνατό

abs()	>>> abs(-3.2) 3.2	Επιστρέφει την απόλυτη τιμή ενός αριθμού
pow(a,b)	>>> pow(3,4) 81	Επιστρέφει τη δύναμη του a υψωμένο στο b
divmod(x,y)	>>> x,y=divmod(15,7); print (x,y) 2 1	Επιστρέφει το πηλίκο και το υπόλοιπο της διαίρεσης x/y
len(L)	>>> len(['u','h','i','c','e','f']) 6	Επιστρέφει το μήκος της λίστας ή του αλφαριθμητικού L

Η Python διαθέτει μια μαθηματική βιβλιοθήκη συναρνήσεων για τις περισσότερες εφαρμογές που μπορεί κανείς να χρειαστεί την οποία πρέπει να εισάγουμε πριν τη χρησιμοποιήσουμε με την εντολή:

```
>>> import math
```

Για να κάνουμε χρήση των συναρνήσεων θα πρέπει να προσδιορίσουμε το όνομα της μονάδας και το όνομα της συνάρτησης χωρισμένα με μια τελεία. Αν θέλετε περισσότερες πληροφορίες για τη βιβλιοθήκη math μπορείτε να πληκτρολογήσετε την εντολή:

```
>>> help(math)
```

Μελετήστε τα παραδείγματα πιο κάτω και τα αποτελέσματα που προκύπτουν.

```
>>> import math
>>> tr=math.sqrt(16)
>>> print(tr)
4.0

>>> print(math.pi)
3.141592653589793

>>> print(math.sin(math.pi/2))
1.0                                     # Το ημίτονο του π/2

>>> print(math.floor(4.9), math.ceil(4.9))
4 5                                   # Στρογγυλοποίηση κάτω και πάνω
```



Ασκήσεις

Χρησιμοποιήστε το περιβάλλον του Thonny για να γράψετε τα προγράμματα που ακολουθούν, να τα αποθηκεύσετε με το αντίστοιχο όνομα και να τα εκτελέσετε (πατώντας Play ή F5) . Όπου είναι χρήσιμο θα σας δίνονται δεδομένα για να τα επαληθεύσετε.

Άσκηση 1 (exer-a01.py)

Να γράψετε πρόγραμμα σε Python που να διαβάζει έναν αριθμό και θα εμφανίζει το διπλάσιό του, το τριπλάσιό του, το αντίθετό του και την απόλυτη τιμή του. Κάθε αποτέλεσμα που θα εμφανίζεται να πλαισιώνεται από το κατάλληλο επεξηγηματικό κείμενο.

**Άσκηση 2 (exer-a02.py)**

Να γράψετε πρόγραμμα σε Python που να διαβάζει την ακτίνα (r) ενός κύκλου και θα υπολογίζει και εμφανίζει την περίμετρό του και το εμβαδόν του.



Παρατήρηση: Οι τύποι για τον υπολογισμό της περιμέτρου και του εμβαδού είναι

$$\text{Περίμετρος} = 2 \cdot \pi \cdot r \text{ και } \text{Εμβαδό} = \pi \cdot r^2.$$

Άσκηση 3 (exer-a03.py)

Δύο σημεία (A και B) πάνω στο επίπεδο ορίζονται από τις συντεταγμένες τους $A(x_1, y_1)$ και $B(x_2, y_2)$. Να γράψετε πρόγραμμα σε Python που να διαβάζει τις συντεταγμένες του A και του B και να υπολογίζει και εμφανίζει την απόστασή τους (d) κάνοντας χρήση του τύπου:



$$d(A, B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Σε όλες τις εισαγωγές δεδομένων φροντίστε να εμφανίζετε τα κατάλληλα διευκρινιστικά μηνύματα

Άσκηση 4 (exer-a04.py)

Να γράψετε πρόγραμμα σε Python που να διαβάζει έναν διψήφιο ακέραιο αριθμό και θα εμφανίζει το άθροισμα των ψηφίων του.

π.χ. αριθμός 39, άθροισμα ψηφίων 12, αριθμός 71, άθροισμα ψηφίων 8



Άσκηση 5 (exer-a05.py)

Επεκτείνετε το προηγούμενο πρόγραμμα για έναν τριψήφιο ακέραιο αριθμό και το αντίστοιχο άθροισμα των ψηφίων του.

π.χ. αριθμός 129, άθροισμα ψηφίων 12, αριθμός 917, άθροισμα ψηφίων 17

