

## Python – 3. Βασικά στοιχεία γλώσσας προγραμματισμού

### Μεταβλητές και τύποι δεδομένων

#### Ονόματα

Τα ονόματα που χρησιμοποιούνται για να ορίσουμε το όνομα της μεταβλητής, συνάρτησης κ.λπ. Μπορεί να αποτελούνται από γράμματα του λατινικού αλφαβήτου (**A - Z, a - z**), τον χαρακτήρα **\_** και τα ψηφία (**0 - 9**), αρκεί ο πρώτος χαρακτήρας να μην είναι ψηφίο και η λέξη να μην είναι δεσμευμένη...

Π.χ. `first_name, code1, x, a1, _a, price_sold`

**Σημείωση:** “δεσμευμένες” είναι οι λέξεις που χρησιμοποιεί η γλώσσα προγραμματισμού στο λεξιλόγιό της π.χ. `print, input`

#### Τύποι δεδομένων

Οι τύποι δεδομένων προσδιορίζουν τον τρόπο παράστασης των δεδομένων εσωτερικά στον υπολογιστή, καθώς και το είδος της επεξεργασίας τους από αυτόν.

Στην Python δε δηλώνουμε ποιο τύπο δεδομένων χρησιμοποιούμε.

Οι χαρακτηριστικοί τύποι δεδομένων στην Python είναι:

- ο **αριθμητικός**,
- ο **λογικός** (`boolean`) και
- οι **συμβολοσειρές** ή αλφαριθμητικά (`strings`).

Οι αριθμοί στην Python είναι κυρίως τριών τύπων:

- **ακέραιοι** (`integer`)
- **αριθμοί κινητής υποδιαστολής** (`floating point`)
- **μγαδικοί αριθμοί** (`complex numbers`),  
τύπος που δε θα μας απασχολήσει στη συνέχεια...

### ΑΚΕΡΑΙΟΙ (`int`)

είναι οι γνωστοί ακέραιοι αριθμοί από τα μαθηματικά.

π.χ. ο αριθμός 3, 198, -4, κ.λπ.

### ΚΙΝΗΤΗΣ ΥΠΟΔΙΑΣΤΟΛΗΣ (`float`)

είναι οι πραγματικοί αριθμοί.

Π.χ. 3.14 και 28.2E-5, όπου το **E** δηλώνει το 10, και το -5 τη δύναμη δηλαδή 28.2 επί 10 εις την -5.

**ΠΡΟΣΟΧΗ!!!**

αντί για κόμμα (,) χρησιμοποιούμε τελεία (.)

### ΛΟΓΙΚΕΣ (`bool`)

είναι ο τύπος που μπορεί να πάρει μόνο δύο τιμές

- **True** (Αληθής) ή
- **False** (Ψευδής).

### ΣΥΜΒΟΛΟΣΕΙΡΕΣ (`str`)

είναι μια ακολουθία χαρακτήρων που ορίζεται με μονά ή διπλά εισαγωγικά, αρκεί με όποια ξεκινάμε με την ίδια να κλείνουμε.

π.χ. "Σήμερα είναι μια ηλιόλουστη μέρα" ή 'Αύριο θα έχει κρύο' ή 'Σταδίου 1125' κ.λπ.

Η ίδια μεταβλητή μπορεί να “περιέχει” διαφορετικού τύπου δεδομένα ανάλογα με τις ανάγκες του προγράμματος

#### Άσκηση

Να γράψετε τον τύπο δεδομένων (`int, float, bool, str`) για καθένα από τα παρακάτω δεδομένα:

19	int	"Γειά"	str
19.0	float	False	bool
27.5E-8	float	'@#\$\$'	str
5E6	float	3.14	float
True	bool	' '	str
'False'	str	-1	int

### Αριθμητικοί τελεστές

Είναι τα σύμβολα που χρησιμοποιούμε για να κάνουμε μαθηματικές πράξεις.

Πράξη	Σύμβολο
πρόσθεση	+
αφαίρεση	-
πολλαπλασιασμός	*
διαίρεση	/
δύναμη	**
ακέραια διαίρεση	// (ή /)
υπόλοιπο ακέραιας διαίρεσης	%

Προτεραιότητα	Πράξεις
1η	**
2η	* / και // %
3η	+ -

#### Ακέραια Διαίρεση

$$\begin{array}{r} 14 \quad | \quad 3 \\ \underline{2} \quad 4 \end{array}$$

Αν διαιρέσουμε το 14 με το 3 παίρνουμε 4 και υπόλοιπο 2!

- Το 4 είναι το αποτέλεσμα της πράξης: **14 / 3** (επειδή 14 και 3 είναι ακέραιοι και η python 2.7 ανάμεσα σε ακεραίους κάνει ακέραια διαίρεση:)
- Το 4 είναι το αποτέλεσμα της πράξης: **14 // 3**
- Το 2 είναι το αποτέλεσμα της πράξης: **14 % 3**

#### Επίσης:

- αν διαιρέσουμε έναν **άρτιο (ζυγό)** με το 2 παίρνουμε υπόλοιπο της ακέραιας διαίρεσης το **0**
- αν διαιρέσουμε έναν **περιττό (μονό)** με το 2 παίρνουμε υπόλοιπο της ακέραιας διαίρεσης το **1**

**Άσκηση**

Να βρείτε το αποτέλεσμα των αριθμητικών πράξεων:

2+8-1	<b>9</b>	5.0/10	<b>0.5</b>
2*9/3	<b>6</b>	0/5	<b>0</b>
2+8*2	<b>18</b>	6+8/2	<b>10</b>
(2+8)*2	<b>20</b>	(6+8)/2	<b>7</b>
45/10	<b>4</b>	10%6	<b>4</b>
45.0/10	<b>4.5</b>	2+10%3	<b>3</b>
45/10.0	<b>4.5</b>	2%4	<b>2</b>
45/10*1.0	<b>4.5</b>	21%10+2**2	<b>5</b>
5/10	<b>0</b>	25-21%2**2	<b>24</b>

**Σχεσιακοί τελεστές**

Χρησιμοποιούνται για τη σύγκριση δύο τιμών ή μεταβλητών, με το αποτέλεσμα μιας σύγκρισης να είναι:

- είτε **True** (Αληθής)
- είτε **False** (Ψευδής)

Περιγραφή	Σύμβολο
μεγαλύτερο	>
μεγαλύτερο ή ίσο	>=
μικρότερο	<
μικρότερο ή ίσο	<=
διάφορο	!=
ίσο	==

**Άσκηση**

Βρείτε το αποτέλεσμα (True ή False) για καθεμία από τις παρακάτω λογικές εκφράσεις:

3==3	True	5>2	True
7>=7	True	2>2	False
5/10>0	False	7%2!=0	True
2!=4	True	3<2	False
5<=5	True	2==2.0	True
5/10.0>0	True	2**0==1	True

**Λογικοί τελεστές**

Στις λογικές πράξεις και εκφράσεις χρησιμοποιούνται οι λογικοί τελεστές not, and, or με τις ακόλουθες λογικές λειτουργίες:

- **not** (όχι): πράξη άρνησης
- **and** (και): πράξη σύζευξης
- **or** (ή): πράξη διάζευξης

Το αποτέλεσμα μιας λογικής πράξης είναι **True** ή **False**.

A	not A
True	False
False	True

A	B	A and B	A or B
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Προτεραιότητα	Πράξη	
1η	<b>not</b>	όχι
2η	<b>and</b>	και
3η	<b>or</b>	ή

**Άσκηση**

Χαρακτηρίστε καθεμία από τις παρακάτω λογικές εκφράσεις ως True ή False.

(8<7) and (5>1)	False
7>=7 and not 5>5	True
(8<7) or (5>1)	True
2==5 and 5!=1 or 8>5	True
(5>2) or (12>11)	True
8>5 or 5>1 and 2==5	True
not (6<=4)	True
(8>5 or 5>1) and 2==5	False

**Βασικές (ενσωματωμένες) συναρτήσεις**

**float()** μετατρέπει ακεραίους και συμβολοσειρές σε δεκαδικούς αριθμούς.

**int()** δέχεται οποιαδήποτε αριθμητική τιμή και τη μετατρέπει σε ακέραιο κόβοντας τα δεκαδικά ψηφία, αν υπάρχουν.

**str()** δέχεται οποιαδήποτε τιμή και την μετατρέπει σε συμβολοσειρά.

**abs()** επιστρέφει την απόλυτη τιμή ενός αριθμού.

**pow(a,b)** επιστρέφει τη δύναμη του a υψωμένη στο b.

**divmod(x,y)** επιστρέφει το ακέραιο πηλίκο και το ακέραιο υπόλοιπο της διαίρεσης x/y.

**input()** χρησιμοποιείται για την εισαγωγή αριθμητικών τιμών από το πληκτρολόγιο και την απόδοσή τους σε μεταβλητή (εισαγωγή).

**raw\_input()** χρησιμοποιείται για την εισαγωγή συμβολοσειρών από το πληκτρολόγιο και την απόδοσή τους σε μεταβλητή (ακατέργαστη εισαγωγή).

**print()** χρησιμοποιείται για την εμφάνιση μηνυμάτων και των τιμών που περιέχουν οι μεταβλητές (εκτύπωση). Στην έκδοση 2.7 την χρησιμοποιούμε χωρίς παρένθεση ...

**Παραδείγματα με συναρτήσεις**

```
>>> a=int(input('Δώσε ακέραιο: '))
Δώσε ακέραιο: 2.7
>>> print a
2
>>> b=float(input('Δώσε πραγματικό: '))
Δώσε πραγματικό: 123
>>> print b
123.0
>>> c=str(a)
>>> d=c+c+c
>>> print d
222
>>> e=-1
>>> f=abs(e)
>>> print f
1
>>> print pow(2,3)
8
>>> a,b=divmod(5,2)
>>> print a,b
2 1
```

**Εντολή εκχώρησης τιμής σε μεταβλητή**

```
x=1
# το x λαμβάνει την τιμή 1
x=x+15
# το x αυξάνει την τιμή του κατά 15
ονομα="Άννα"
# το ονομα λαμβάνει την τιμή Άννα
m1=m2=m3=23
# τα m1,m2,m3 λαμβάνουν την τιμή 23
x,y=10,'X'
# το x λαμβάνει την τιμή 10 και το y την τιμή X
x,y,z=3,7,18
# τα x,y,z λαμβάνουν τις τιμές 3,7,18 αντίστοιχα
x+=1
# το x αυξάνει την τιμή του κατά 1 - Ισοδύναμο του x = x+1
y-=10
# το y μειώνεται κατά 15 - Ισοδύναμο: y = y – 10
```

**Δομή Προγράμματος και Καλές Πρακτικές****1η “καλή πρακτική”**

Να δίνουμε ένα χαρακτηριστικό **τίτλο** στο πρόγραμμα με τη μορφή σχολίων, τα οποία ξεκινάνε με το σύμβολο #.

Αυτό, παρότι δεν είναι αναγκαίο, είναι ιδιαίτερα χρήσιμο.

**2η “καλή πρακτική”**

Να προσέχουμε τα **κενά διαστήματα** πριν την κάθε εντολή καθώς η Python βασίζεται σε αυτά, για να ορίσει ομάδες εντολών.

**3η “καλή πρακτική”**

Να επιλέγουμε τους κατάλληλους **τελεστές**.

**4η “καλή πρακτική”**

Να χρησιμοποιούμε τα **ίδια εισαγωγικά** (μονά εισαγωγικά με μονά, διπλά εισαγωγικά με διπλά) για μια συμβολοσειρά.

**5η “καλή πρακτική”**

Να προσθέτουμε μέσα στον κώδικα **επεξηγηματικά σχόλια**. Τα σχόλια που περιγράφουν τη λειτουργία ενός προγράμματος, το ρόλο των μεταβλητών και τη λειτουργία πολύπλοκων τμημάτων κώδικα, αποτελούν παράδειγμα καλού προγραμματιστικού στυλ.

Θυμίζουμε ότι τα σχόλια περιγράφουν τη λειτουργία ενός προγράμματος ή γενικότερα, ενός τμήματος κώδικα και γράφονται, για να βοηθήσουν τους ανθρώπους -και όχι τον υπολογιστή- στην κατανόηση και συντήρηση ενός προγράμματος.

Όταν ένα πρόγραμμα μεταφράζεται, τα σχόλια αγνοούνται.

**6η “καλή πρακτική”**

Να δίνουμε **ονόματα μεταβλητών** που έχουν σχέση με τη χρήση τους.