

3.1 Μεταβλητές και τύποι δεδομένων

3.1.1 Τύποι δεδομένων

Ένα πρόγραμμα συνήθως επεξεργάζεται δεδομένα τα οποία μπορεί να είναι αποθηκευμένα στην κύρια μνήμη του υπολογιστή, σε αποθηκευτικό μέσο ή η εισαγωγή τους να γίνεται από κάποια μονάδα. 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)
- μιγαδικοί αριθμοί (complex numbers), τύπος που δε θα μας απασχολήσει στη συνέχεια.

Παραδείγματα

Ο 19, αποτελεί παράδειγμα ακεράιου.

Οι 256.14 και 28.2E-5, όπου το σύμβολο E δηλώνει δύναμη του 10, είναι παραδείγματα αριθμών κινητής υποδιαστολής (ή floats για συντομία). Σε αυτή την περίπτωση, το 28.2E-5 σημαίνει $28.2 * 10^{-5}$. Παρατηρούμε ότι το δεκαδικό τμήμα διαχωρίζεται με το χαρακτήρα τελεία "." και όχι το κόμμα ",",.

Ο λογικός τύπος (boolean) που δέχεται μόνο δύο τιμές, την τιμή True (Αληθής) και την τιμή False (Ψευδής) και έχει σκοπό την καταγραφή του αποτελέσματος ενός ελέγχου.

Οι συμβολοσειρές είναι μια ακολουθία από χαρακτήρες. Μια συμβολοσειρά μπορεί να αποτελείται από περισσότερες από μία λέξεις. Οι λέξεις μπορούν να είναι σε κάθε γλώσσα που υποστηρίζεται από το πρότυπο Unicode, άρα σε ελληνική, αγγλική κ.ο.κ. Μπορούμε να ορίσουμε μια συμβολοσειρά με μονά εισαγωγικά ή με διπλά, αλλά όχι ανάμικτα. Με ότι ξεκινάμε, θα πρέπει πάλι να κλείνουμε.

Παράδειγμα: "221051445" ή 'Καλημέρα'.

Για να ελέγξουμε τον τύπο δεδομένων χρησιμοποιούμε την εντολή type ().

Παράδειγμα

```
>>> type( 1 )
<type 'int'>
>>> type( 3.14 )
<type 'float'>
>>> a = True
>>> type(False )
<type 'bool'>
>>> type( 'Αρετή' )
<type 'str'>
```

3.2 Αριθμητικές και λογικές πράξεις και εκφράσεις

Χρησιμοποιώντας τιμές κάθε τύπου δεδομένων, μπορούμε να κάνουμε διάφορες πράξεις, χρησιμοποιώντας τους αντίστοιχους τελεστές. Οι τελεστές (operators) είναι σύμβολα ή λέξεις για τη δημιουργία αριθμητικών και λογικών εκφράσεων. Οι βασικότεροι τελεστές στη γλώσσα Python είναι:

31

Μέρος 1. Εμβάθυνση σε βασικές έννοιες

Αριθμητικοί τελεστές: Είναι τα σύμβολα που χρησιμοποιούμε για να κάνουμε μαθηματικές πράξεις. Στη γλώσσα Python χρησιμοποιούμε τους παρακάτω βασικούς αριθμητικούς τελεστές:

Πρόσθεση +

Αφαίρεση -

Πολλαπλασιασμός *

Διαίρεση /

Ύψωση σε δύναμη **

Το υπόλοιπο της ακέραιας διαίρεσης %

Σε κάθε έκφραση στην οποία υπάρχουν αριθμητικοί τελεστές ακολουθείται μια προσδιορισμένη ιεραρχία πράξεων, που είναι:

1. Ύψωση σε δύναμη.

2. Πολλαπλασιασμός, διαίρεση, υπόλοιπο ακέραιας διαίρεσης.

3. Πρόσθεση, αφαίρεση.

Αν θέλουμε να αλλάξουμε την ιεραρχία των πράξεων, μπορούμε να χρησιμοποιήσουμε παρενθέσεις. Για παράδειγμα, στην έκφραση $(2+3)*5$ θα εκτελεστεί πρώτα η πρόσθεση μέσα στην παρένθεση και μετά, το αποτέλεσμα θα πολλαπλασιαστεί επί 5, σε αντίθεση με την έκφραση $2+3*5$ στην οποία, πρώτα θα γίνει ο πολλαπλασιασμός και μετά η πρόσθεση.

Σχεσιακοί (ή συγκριτικοί) τελεστές: χρησιμοποιούνται για τη σύγκριση δύο τιμών ή μεταβλητών, με το αποτέλεσμα μιας σύγκρισης να είναι είτε True (Αλη-

θής) είτε False (Ψευδής). Στη γλώσσα Python χρησιμοποιούνται οι παρακάτω βασικοί σχεσιακοί τελεστές:
Μικρότερο από <
Μικρότερο ή ίσο από <=
Μεγαλύτερο από >
Μεγαλύτερο ή ίσο από >=
Ίσο με ==
Διάφορο από !=

Τελεστές λογικών πράξεων: Στις λογικές πράξεις και εκφράσεις χρησιμοποιούνται οι λογικοί τελεστές not (ΟΧΙ), and (ΚΑΙ), or (Η) με τις ακόλουθες λογικές λειτουργίες:

- not (ΟΧΙ): πράξη άρνησης
- and (ΚΑΙ): πράξη σύζευξης
- or (Η): πράξη διάζευξης.

Το αποτέλεσμα μιας λογικής πράξης είναι True (Αληθής) ή False (Ψευδής) σύμφωνα με τον παρακάτω πίνακα:

P	Q	P and Q	P or Q	Not P
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Η προτεραιότητα των λογικών τελεστών είναι not, and, or με αυτή τη σειρά. Πίνακας 3-1. Παραδείγματα αριθμητικών/λογικών εκφράσεων

Αριθμητικές πράξεις Αποτελέσματα

```
>>> 2+8*2 18
>>> 2**2+4/2-3*4 -6
>>> 45 / 10
4.4 # Στην Python, η διαίρεση ακεραίων αριθμών μας επιστρέφει το ακέραιο πηλίκο
>>> 45 % 10 5 # μας επιστρέφει το υπόλοιπο της ακεραίας διαίρεσης 45 / 10
>>> 45.0 /10
4.5 # Η διαίρεση αριθμών κινητής υποδιαστολής, μας επιστρέφει το πηλίκο, ως αριθμό κινητής υποδιαστολής
```

Μέρος 1. Εμβάθυνση σε βασικές έννοιες

Λογικές εκφράσεις με συγκριτικούς τελεστές

Αποτελέσματα

```
>>> 23 == 23 True
>>> 34 != 45 True
>>> 56 <= 12 False
```

Λογικές εκφράσεις με τελεστές λογικών πράξεων Αποτελέσματα

```
>>> (12<11) and (23>10) False
>>> (12<11) or (23>10) True
>>> not(56<=12) True
```

Μεταβλητές

Η γλώσσα Python παρέχει εντυπωσιακές εναλλακτικές δυνατότητες για τη διαχείριση μεταβλητών που διευκολύνουν τον προγραμματιστή. Για τη χρησιμοποίηση μιας μεταβλητής δεν απαιτείται η δήλωσή της, ενώ μπορεί να εκχωρήσουμε διαφορετικούς τύπους τιμών σε μια μεταβλητή κατά τη διάρκεια ενός προγράμματος, όπως ακέραιες τιμές, κινητής υποδιαστολής και συμβολοσειρές.

Για να χρησιμοποιήσουμε μια μεταβλητή απαιτείται να της δώσουμε ένα όνομα και στη συνέχεια να της εκχωρήσουμε κάποια τιμή. Στη γλώσσα Python υπάρχουν ορισμένοι κανόνες που πρέπει να ακολουθούμε σχετικά με το όνομα μιας μεταβλητής. Έτσι, για παράδειγμα, δεν επιτρέπεται να ξεκινά το όνομα μιας μεταβλητής με αριθμό και το όνομα που θα δώσουμε δεν πρέπει να είναι όμοιο με κάποιο όνομα ενσωματωμένης συνάρτησης ή εντολής. Συνηθίζουμε να δίνουμε ένα όνομα σχετικό με το είδος της μεταβλητής με λατινικούς χαρακτήρες, που

34 Προγραμματισμός Υπολογιστών
μπορεί να συνοδεύεται από κάποιον αριθμό, όπως για παράδειγμα: opoma1, opoma_2, timi, mesi_timi, embado κ.ά.

Για να εκχωρήσουμε μια τιμή σε μια μεταβλητή, χρησιμοποιούμε τον τελεστή

ίσον "=" , όπως για παράδειγμα $a = 125$.

Στο αριστερό μέρος δίνουμε το όνομα της μεταβλητής, στη συνέχεια χρησιμοποιούμε τον τελεστή εκχώρησης "=" και στο δεξιό μέρος βάζουμε την τιμή, μια άλλη μεταβλητή ή μια έκφραση που έχει ως αποτέλεσμα μια τιμή.

Προσοχή! ο τελεστής "=" δεν έχει τη σημασία που έχει το σύμβολο της ισότητας, όπως το χρησιμοποιούμε στα μαθηματικά. Στις περισσότερες γλώσσες προγραμματισμού, ο τελεστής εκχώρησης τιμής "=" σημαίνει ότι εκχωρούμε στη μεταβλητή μια τιμή ενός τύπου δεδομένου και ταυτόχρονα την εισάγουμε στη θέση μνήμης που αντιστοιχεί στη μεταβλητή με αυτό το όνομα. Για παράδειγμα, η εντολή $a = 125$ σημαίνει ότι η μεταβλητή με όνομα a αντιστοιχεί σε μια θέση μνήμης του υπολογιστή και η ακέραια τιμή 125 εισάγεται στη θέση αυτή. Αν στη συνέχεια θέσουμε $a = 250$, στη θέση μνήμης που αναφέρεται η μεταβλητή a εισάγεται η νέα ακέραια τιμή 250 και η παλαιά τιμή 125 παύει πλέον να υπάρχει, αφού έχει αντικατασταθεί από τη 250.

Όλα τα δεδομένα σε ένα πρόγραμμα Python αναπαρίστανται με αντικείμενα ή με σχέσεις μεταξύ των αντικειμένων, με κάθε αντικείμενο να έχει μια ταυτότητα

```
>>> x = 4.5
>>> y = x
>>> y = y + 3
>>> x
4.5
>>> y
7.5
x
x
4.5
x 4.5
y 7.5
```

Προσοχή! ο τελεστής "=" δεν έχει τη σημασία που έχει το σύμβολο της ισότητας, όπως το χρησιμοποιούμε στα μαθηματικά. Στις περισσότερες γλώσσες προγραμματισμού, ο τελεστής εκχώρησης τιμής "=" σημαίνει ότι εκχωρούμε στη μεταβλητή μια τιμή ενός τύπου δεδομένου και ταυτόχρονα την εισάγουμε στη θέση μνήμης που αντιστοιχεί στη μεταβλητή με αυτό το όνομα. Για παράδειγμα, η εντολή $a = 125$ σημαίνει ότι η μεταβλητή με όνομα a αντιστοιχεί σε μια θέση μνήμης του υπολογιστή και η ακέραια τιμή 125 εισάγεται στη θέση αυτή. Αν στη συνέχεια θέσουμε $a = 250$, στη θέση μνήμης που αναφέρεται η μεταβλητή a εισάγεται η νέα ακέραια τιμή 250 και η παλαιά τιμή 125 παύει πλέον να υπάρχει, αφού έχει αντικατασταθεί από τη 250.

Όλα τα δεδομένα σε ένα πρόγραμμα Python αναπαρίστανται με αντικείμενα ή με σχέσεις μεταξύ των αντικειμένων, με κάθε αντικείμενο να έχει μια ταυτότητα

(identity), έναν τύπο και μία τιμή. Για παράδειγμα, το 12 είναι ένα αντικείμενο με τιμή 12, τύπου int (ακέραιος).

$a = 1$ $a = 2$ $a = b = 2$

Οι μεταβλητές στην Python λειτουργούν σαν ετικέτες με κάποιο όνομα, που χρησιμεύουν για να αναφερόμαστε (ή αλλιώς για να δείχνουμε) σε κάποια αντικείμενα. Ο τύπος της μεταβλητής είναι ο εκάστοτε τύπος του αντικειμένου στην οποία αναφέρεται. Στο παράδειγμα $a = 125$, δημιουργείται η μεταβλητή με όνομα a και αναφέρεται στο αντικείμενο, με τιμή 125, με ακέραιο τύπο δεδομένων. Το σύμβολο "=" δημιουργεί ένα είδος δεσίματος μεταξύ του αντικειμένου 125 και της μεταβλητής με όνομα a . Ο τύπος της μεταβλητής είναι και αυτός ακέραιος, αφού αναφέρεται σε αντικείμενο με ακέραιο τύπο δεδομένων.

Όταν θέσουμε στη συνέχεια $a = 250.0$, η μεταβλητή a παύει πλέον να δείχνει το αντικείμενο με τιμή 125 και δημιουργείται ένα νέο "δέσιμο", ώστε να αναφέρεται στο αντικείμενο με τιμή 250.0 (κινητής υποδιαστολής τύπος δεδομένων). Ο τύπος δεδομένων της μεταβλητής a τώρα έχει αλλάξει και είναι κινητής υποδιαστολής, όμοιος με τον τύπο του νέου αντικειμένου στο οποίο αναφέρεται.

Η Python παρακολουθεί όλες τις τιμές και τις διαγράφει όταν πάψουν να υπάρχουν μεταβλητές που να αναφέρονται σε αυτές. Η διαδικασία αυτή ονομάζεται συλλογή σκουπιδιών (garbage collection).

Στο παράδειγμα που ακολουθεί η μεταβλητή `a` παίρνει ως τιμή διαδοχικά ένα ακέραιο, ένα πραγματικό, μια λογική τιμή και τέλος μια συμβολοσειρά. Κάθε φορά ελέγχεται ο τύπος δεδομένων με την εντολή `type`.

```
>>> a = 125
>>> type( a )
<type 'int'>
>>> a = 250.7
>>> type( a )
<type 'float'>
>>> a = True
>>> type( a )
<type 'bool'>
>>> a = 'Καλημέρα'
>>> type( a )
<type 'str'>
```

Αν θέλουμε να εμφανίσουμε το περιεχόμενο της μεταβλητής, τότε μπορούμε να χρησιμοποιήσουμε την εντολή `print` μαζί με το όνομα της μεταβλητής. Για παράδειγμα: `message = 'Καλημέρα'`, η εντολή `print message` θα εμφανίσει στην οθόνη τη λέξη Καλημέρα. Στο όνομα της μεταβλητής `message`, που ακολουθεί μετά την `print`, δεν πρέπει να χρησιμοποιηθούν εισαγωγικά (μονά ή διπλά), διότι τότε θα εμφανιστεί στην οθόνη η συμβολοσειρά `message`. Παρόμοια, αν θέσουμε `a = 1234`, η `print a` θα εμφανίσει την τιμή 1234 στην οθόνη.

Βασικές εντολές στη γλώσσα Python

Στις προηγούμενες παραγράφους παρουσιάστηκαν μερικές από τις βασικές εντολές (statements) στην Python, όπως:

`print`: χρησιμοποιείται για την εμφάνιση τιμών στην οθόνη του υπολογιστή, με ποικίλες μορφές, όπως:

- `print` όνομα_μεταβλητής, π.χ. `print name`
- `print` αριθμός, π.χ. `print 1045.34`
- `print` 'συμβολοσειρά', π.χ. `print 'θαλασσινός αέρας'`
- `print` όνομα_μεταβλητής (τελεστής) αριθμός

```
>>> name = 'kostas'
>>> print name
Kostas
>>> print 1045.34
1045.34
>>> message = "Τώρα "
>>> print message*3
Τώρα Τώρα Τώρα
```

Εκχώρηση τιμής σε μια μεταβλητή: χρησιμοποιείται το σύμβολο `=` για την εκχώρηση μιας τιμής σε μια μεταβλητή.

όνομα_μεταβλητής = τιμή μεταβλητής

```
tree = 'Πεύκο'
```

```
x = 135
```

```
x = x**2 + 34
```

Εισαγωγή τιμής σε μια μεταβλητή από το πληκτρολόγιο. Το πρόγραμμα αναμένει από το χρήστη να πληκτρολογήσει μια τιμή. Η τιμή που εισάγεται αποδίδεται σε μια μεταβλητή. Προς τούτο χρησιμοποιούνται δύο εντολές: η `input()` και η `raw_input()`.

Σύνταξη: Όνομα μεταβλητής = `input('μήνυμα')`.

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

```
Δώσε έναν αριθμό: 34
```

```
>>> print number
```

```
34
```

Αν θέλουμε να εισάγουμε ένα αλφαριθμητικό (ή αριθμό με τη μορφή αλφαριθμητικού), χρησιμοποιούμε τη `raw_input`:

Σύνταξη: Όνομα μεταβλητής = `raw_input ()`

38 Προγραμματισμός Υπολογιστών

```
>>> name = raw_input('Δώσε το όνομά σου :')
```

```
Δώσε το όνομά σου : Kostas
```

```

>>> print name
Kostas
>>> tree = raw_input('Δώσε το όνομα ενός δέντρου με άνθη που βλέπεις
πηγαίνοντας στο σχολείο: ')
Δώσε το όνομα ενός δέντρου με άνθη που βλέπεις πηγαίνοντας στο σχο-
λείο: Αμυγδαλιά
>>> print 'Το δέντρο ',tree,'είναι όμορφο ανθισμένο'
Το δέντρο Αμυγδαλιά είναι όμορφο ανθισμένο
Γενικά, ότι εισάγεται με τη raw_input θεωρείται αυτόματα αλφαριθμητικό, ενώ η
input προσπαθεί να το υπολογίσει. Για παράδειγμα, αν δώσουμε στην input το
όνομα μιας μεταβλητής, θα επιστρέψει το περιεχόμενο της μεταβλητής.
Εισαγωγή σχολίων
Τα σχόλια σε ένα πρόγραμμα διευκολύνουν την κατανόησή του. Στην Python,
τα σχόλια εισάγονται θέτοντας μπροστά από αυτά το σύμβολο # . Τα σχόλια
μπορούν να αρχίζουν και μετά από εντολές στη μέση μιας γραμμής. Ό,τι βρί-
σκεται δεξιά από το #, αγνοείται από το διερμηνευτή.
# Πρόγραμμα πολλαπλασιασμός δύο αριθμών
>>> x=input('Δώσε τον πρώτο αριθμό: ')
Δώσε το πρώτο αριθμό: 34
>>> y=input('Δώσε το δεύτερο αριθμό: ')
Δώσε το δεύτερο αριθμό: 2
>>> ginomeno = x*y # πολλαπλασιάζει τις τιμές των x και y και
# το αποτέλεσμα το εκχωρεί στη μεταβλητή ginomeno
>>> print ginomeno
68

```

Μέρος 1. Εμβάθυνση σε βασικές έννοιες

3.3 Βασικές (ενσωματωμένες) συναρτήσεις

Η Python παρέχει μια ποικιλία ενσωματωμένων συναρτήσεων για τη μετατροπή τιμών δεδομένων από έναν τύπο σε έναν άλλο, όπως οι: int(), float() και str().

Δραστηριότητα εμπέδωσης

Στο περιβάλλον της γλώσσας Python, επαληθεύστε τις παρακάτω συναρτήσεις:

- Η float() μετατρέπει ακέραιους και συμβολοσειρές σε δεκαδικούς αριθμούς.
- Η int() δέχεται οποιαδήποτε αριθμητική τιμή και τη μετατρέπει σε ακέραιο κόβοντας τα δεκαδικά ψηφία, αν υπάρχουν.
- Η str() δέχεται οποιαδήποτε τιμή και την μετατρέπει σε συμβολοσειρά.
- Η abs() επιστρέφει την απόλυτη τιμή ενός αριθμού.
- Η pow(a,b) επιστρέφει τη δύναμη του α υψωμένη στο β.
- Η divmod(x,y) επιστρέφει το ακέραιο πηλίκο και το ακέραιο υπόλοιπο της διαίρεσης x/y.

```

>>> float(10)
10.0
>>> int(5.678)
5
>>> str(5.678)
5.678
>>> abs(-45)
45
>>> divmod(10,3)
(3, 1)
>>> pow(2,3)
8

```

Αν θέλουμε να διαβάσουμε από το πληκτρολόγιο έναν ακέραιο αριθμό με τη συνάρτηση input(), πρέπει να χρησιμοποιούμε και τη συνάρτηση int()

```
a=int(input('Δώσε έναν αριθμό :'))
```

```
40
```

Παράδειγμα

```

>>> import math
>>> a=int(input('Δώσε έναν ακέραιο αριθμό: '))
Δώσε έναν ακέραιο αριθμό: 2345.10
# ο χρήστης δίνει την τιμή 2345.10
>>> print a
2345

```

εμφανίζεται η ακέραια τιμή του αριθμού αποκόπτοντας τα δεκαδικά ψηφία

Δραστηριότητα

Συμβουλευτείτε τη «βοήθεια» του προγραμματιστικού περιβάλλοντος και πειραματιστείτε με τις ενσωματωμένες συναρτήσεις που παρέχει η γλώσσα Python.

Εξωτερικές βιβλιοθήκες

Εκτός από τις ενσωματωμένες συναρτήσεις οι οποίες περιλαμβάνονται στη γλώσσα Python, μπορεί κανείς να βρει πληθώρα εξωτερικών βιβλιοθηκών (αρθρώματα ή modules) στους Διαδικτυακούς τόπους υποστήριξης της γλώσσας. Μια βιβλιοθήκη είναι ένα αρχείο το οποίο περιέχει μια συλλογή από σχετικές συναρτήσεις.

Ενδεικτικά, μπορούν να αναφερθούν: η Μαθηματική βιβλιοθήκη και οι βιβλιοθήκες γραφικών. Οι βιβλιοθήκες αυτές για να χρησιμοποιηθούν, θα πρέπει πρώτα να εισαχθούν στο πρόγραμμά μας. Η εισαγωγή αυτή γίνεται με την εντολή `import`. Για παράδειγμα, προκειμένου να χρησιμοποιήσουμε μαθηματικές συναρτήσεις σε ένα πρόγραμμα, θα πρέπει μία από τις αρχικές εντολές του προγράμματός μας να είναι: `import math`.

Για να έχουμε πρόσβαση σε μια από τις συναρτήσεις, θα πρέπει να δηλώσουμε το όνομα της μονάδας και το όνομα της συνάρτησης, χωρισμένα με μια τελεία, μορφή που ονομάζεται συμβολισμός με τελεία (dot notation). Ας δούμε ένα παράδειγμα συνάρτησης για την τετραγωνική ρίζα, `sqrt()`.

Μέρος 1. Εμβάθυνση σε βασικές έννοιες

```
>>> import math
>>> riza = math.sqrt(2)
>>> print riza
1.41421356237
>>> math.sqrt(3)
1.7320508075688772
>>> x = math.pi
>>> print x
3.1415926535
```