

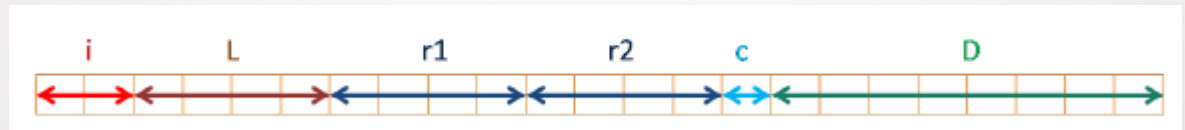
Python: Μεταβλητές, Εκφράσεις, Είσοδος & Έξοδος

Ενότητες

- Μεταβλητές
- Μεταβλητές και Τύποι Δεδομένων
- Εκφράσεις
- Λογικές Εκφράσεις
- Είσοδος & Έξοδος
- Σχόλια
- Χαρακτήρες Διαφυγής

Μεταβλητές

- Η **μεταβλητή**, λοιπόν, είναι ένα συμβολικό όνομα μιας περιοχής της μνήμης στην οποία μπορούμε να γράψουμε και να ανακτήσουμε δεδομένα μέσω του συμβολικού αυτού ονόματος



- Η πράξη με την οποία αναθέτουμε τιμές στις μεταβλητές λέγεται **εκχώρηση**
- Τις μεταβλητές μπορούμε να τις χρησιμοποιήσουμε μέσα σε **εκφράσεις**

Μεταβλητές και Τύποι Δεδομένων

- Οι βασικοί **τύποι δεδομένων** των **μεταβλητών** της Python είναι οι ακόλουθοι:
 - αριθμητικές (**numerical**) μεταβλητές
 - λογικές (**boolean**) μεταβλητές
 - αλφαριθμητικά (**strings**)
 - λίστες (**lists**)
 - πλειάδες (**tuples**)
 - λεξικά (**dictionaries**)

Μεταβλητές και Τύποι Δεδομένων

- Οι αριθμητικές μεταβλητές χωρίζονται σε:
 - **ακέριες μεταβλητές (int)**
 - **μεταβλητές ακέραιες υψηλής ακρίβειας (float)**
 - **πραγματικές μεταβλητές**
 - **μιγαδικές μεταβλητές**

Μεταβλητές και Τύποι Δεδομένων

- Παράδειγμα

```
>>> import sys
>>> a=1
>>> a
1
>>> type(a)
<class 'int'>
>>> sys.getsizeof(a)
28
>>>
```

Εκχώρηση τιμής
(αρχικοποίηση μεταβλητής)

Ζητάμε την προβολή της
τιμής της μεταβλητής

Ζητάμε πληροφόρηση για
τον τύπο της μεταβλητής

Ζητάμε πληροφόρηση για το
μέγεθος (χώρο που
καταλαμβάνει στη μνήμη) της
μεταβλητής – Απαιτείται η
χρήση της βιβλιοθήκης sys

Μεταβλητές και Τύποι Δεδομένων

- Παραδείγματα

Παρατηρήστε την εύκολη αλλαγή στον τύπο δεδομένων αλλάζοντας απλά την τιμή

```
>>> a=10000000000
>>> type(a)
<class 'int'>
>>> sys.getsizeof(a)
32
>>> a=3.14
>>> type(a)
<class 'float'>
>>> sys.getsizeof(a)
24
>>> a=300000000000.14
>>> type(a)
<class 'float'>
>>> sys.getsizeof(a)
24
>>>
```

```
>>> a='1'
>>> type(a)
<class 'str'>
>>> sys.getsizeof(a)
50
>>> a='12'
>>> sys.getsizeof(a)
51
>>> a='123'
>>> sys.getsizeof(a)
52
>>>
```

Μεταβλητές και Τύποι Δεδομένων

- Παράδειγμα συμβολοσειρών (strings)

```
>>> a='1'  
>>> type(a)  
<class 'str'>  
>>> sys.getsizeof(a)  
50  
>>> a='12'  
>>> sys.getsizeof(a)  
51  
>>> a='123'  
>>> sys.getsizeof(a)  
52  
>>>
```

Τι παρατηρείτε;

Μεταβλητές και Τύποι Δεδομένων

- Τα **αλφαριθμητικά** είναι μεταβλητές οι οποίες έχουν τη δυνατότητα να αποθηκεύσουν συμβολοσειρές που μπορεί να αποτελούνται από γράμματα, αριθμούς ή και άλλα σύμβολα
- Τις συμβολοσειρές τις περικλείουμε μέσα σε εισαγωγικά (μονά ή διπλά) ώστε να ξεχωρίσουν από οτιδήποτε άλλο. Για παράδειγμα, το **'test'** (το οποίο είναι ακριβώς το ίδιο με το **"test"**) είναι μία συμβολοσειρά που αποτελείται από το χαρακτήρα **t**, στη συνέχεια τον χαρακτήρα **e**, στην συνέχεια τον χαρακτήρα **s** και τελειώνει με τον χαρακτήρα **t**

Μεταβλητές και Τύποι Δεδομένων

- Οι **λογικές μεταβλητές (boolean)** είναι μεταβλητές οι οποίες παίρνουν δύο τιμές **True** και **False**
- Έτσι, είναι δυνατόν να ορίσουμε απευθείας **x=True** ή να γίνει αυτό μέσα από μία λογική σύγκριση
- π.χ. **x=a<6**, όπου το **x** θα πάρει την τιμή **True** αν πράγματι το **a** είναι μικρότερο του 6 και **False** σε κάθε άλλη περίπτωση

Μεταβλητές και Τύποι Δεδομένων

- Παράδειγμα λογικών μεταβλητών (boolean)

```
>>> b=True
>>> type(b)
<class 'bool'>
>>> a=12
>>> x=a<6
>>> x
False
```

Εκφράσεις

- Οι **εκφράσεις** μιας γλώσσας είναι συνδυασμός **τελεστών** και **μεταβλητών**, η αποτίμηση των οποίων δημιουργεί μία νέα τιμή η οποία μπορεί να εκχωρηθεί σε μία μεταβλητή ή να χρησιμοποιηθεί σαν μεταβλητή

Εκφράσεις

- Οι σημαντικότεροι τελεστές που εφαρμόζονται πάνω σε αριθμητικές μεταβλητές είναι οι ακόλουθοι:
 - μοναδιαίοι: +, - (πρόσημα)
 - προσθετικοί +, -
 - πολλαπλασιαστικοί *, /, // (// ακέραια διαίρεση)
 - διάφοροι άλλοι όπως
 - % υπόλοιπο διαίρεσης
 - ** ύψωση σε δύναμη

Εκφράσεις

- Για την **προτεραιότητα των πράξεων** ακολουθείται η συνήθης πρακτική
 - Μεγαλύτερη προτεραιότητα έχουν οι τελεστές υπολοίπου **%** και ύψωσης σε δύναμη ******
 - Μετά ακολουθούν οι πολλαπλασιαστικοί τελεστές
 - Τέλος, με μικρότερη προτεραιότητα έχουμε τους προσθετικούς (συμπεριλαμβανομένων των μοναδιαίων, δηλαδή των προσήμων)
 - Η αποτίμηση των τελεστών με την ίδια προτεραιότητα γίνεται από τα αριστερά στα δεξιά (έχει σημασία αυτό στην ακέραια διαίρεση)

Εκφράσεις

- Παράδειγμα προτεραιότητα των πράξεων

```
>>> a=3.16
>>> b=2.1
>>> c=3*(a+b)**2+5
>>> c
88.0028
>>> 3/5
0.6
>>> 3//5
0
>>> 5//2*3
6
>>> 5*2//3
3
```

Εκφράσεις

- Οι τελεστές ανάλογα με τις μεταβλητές πάνω στις οποίες εφαρμόζονται αποκτούν και διαφορετική σημασία.
- Κάποιος τελεστής μπορεί ακόμα να έχει νόημα να εφαρμοστεί σε μία μεταβλητή κάποιου τύπου αλλά να μην έχει νόημα να εφαρμοστεί στις μεταβλητές άλλου τύπου.

Εκφράσεις

- Παράδειγμα ρόλου τελεστών

```
>>> s1='1234'  
>>> s2='abc'  
>>> s3=s1+2*s2+'!!!'  
>>> s3  
'1234abcabc!!!'  
>>>
```

Τι παρατηρείτε για τους
τελεστές + και *;

Λογικές Εκφράσεις

- Μία **λογική μεταβλητή** μπορεί να πάρει τις τιμές **True** (σημαίνει **αληθής**, ότι η λογική παράσταση που αποτιμήθηκε σε αυτό ισχύει) ή **False** (σημαίνει **ψευδής**, ότι η λογική παράσταση που αποτιμήθηκε σε αυτό δεν ισχύει)
- Μία λογική έκφραση είναι, σε αναλογία με την αριθμητική έκφραση, ένας συνδυασμός λογικών μεταβλητών και λογικών τελεστών

Λογικές Εκφράσεις

- Οι λογικοί τελεστές είναι οι ακόλουθοι:
 - ==: ισότητα
 - <: μικρότερο
 - >: μεγαλύτερο
 - <=: μικρότερο ή ίσο
 - >=: μεγαλύτερο ή ίσο
 - !=: διάφορο
 - **and** λογική σύζευξη, **or** λογική διάζευξη, **not** λογική άρνηση
- Ισχύουν και πάλι κανόνες προτεραιότητας με το **not** να έχει τη μεγαλύτερη, το **and** να ακολουθεί και το **or** να έχει τη μικρότερη

Λογικές Εκφράσεις

- Παραδείγματα λογικών εκφράσεων
 - $A > B$ είναι μία λογική έκφραση που παίρνει την τιμή **True** αν το **A** είναι μεγαλύτερο του **B** και **False** σε κάθε άλλη περίπτωση
 - $\text{not } A == \text{True}$ εάν $A == \text{False}$
 - $A \text{ and } B == \text{True}$ εάν $A == \text{True}$ και $B == \text{True}$
 - $A \text{ or } B == \text{True}$ εάν $A == \text{True}$ ή $B == \text{True}$
 - $A < 4 \text{ and } B > 3$ είναι αληθές όταν και το **A** είναι μικρότερο του 4 αλλά και το **B** μεγαλύτερο του 3
 - $\text{not}(A < 4 \text{ or } B > 3)$ η έκφραση μέσα στην παρένθεση είναι αληθής όταν ή το **A** είναι μικρότερο του 4 ή το **B** μεγαλύτερο του 3

Είσοδος και Έξοδος

- **Είσοδο δεδομένων** έχουμε όταν ο υπολογιστής ζητάει από τον χρήστη να του δώσει τιμή για κάποια μεταβλητή
- **Έξοδο δεδομένων** έχουμε όταν ο υπολογιστής πληροφορεί για κάτι τον χρήστη

Είσοδος και Έξοδος

- Στην Python η είσοδος δεδομένων από το πληκτρολόγιο γίνεται με την `input()`

```
x=input('Δώστε τιμή για το x: ')
```

Είσοδος και Έξοδος

- Παράδειγμα

```
>>> x=input('Give me the value of x: ')
Give me the value of x: 34
>>> x
'34'
>>> 2*x
'3434'
>>> x=int(input('Give me the value of x: '))
Give me the value of x: 34
>>> x
34
>>> 2*x
68
>>> x=int(input('Give me the value of x: '))
Give me the value of x: 3.4
Traceback (most recent call last):
  File "<pyshell#23>", line 1, in <module>
    x=int(input('Give me the value of x: '))
ValueError: invalid literal for int() with base 10: '3.4'
>>>
```

Είσοδος και Έξοδος

- Στην Python η έξοδος (εμφάνιση τιμών μεταβλητών στην οθόνη) γίνεται με την **print()**

```
print("Η τιμή του x είναι: ",x)
```


Είσοδος και Έξοδος

- Παράδειγμα

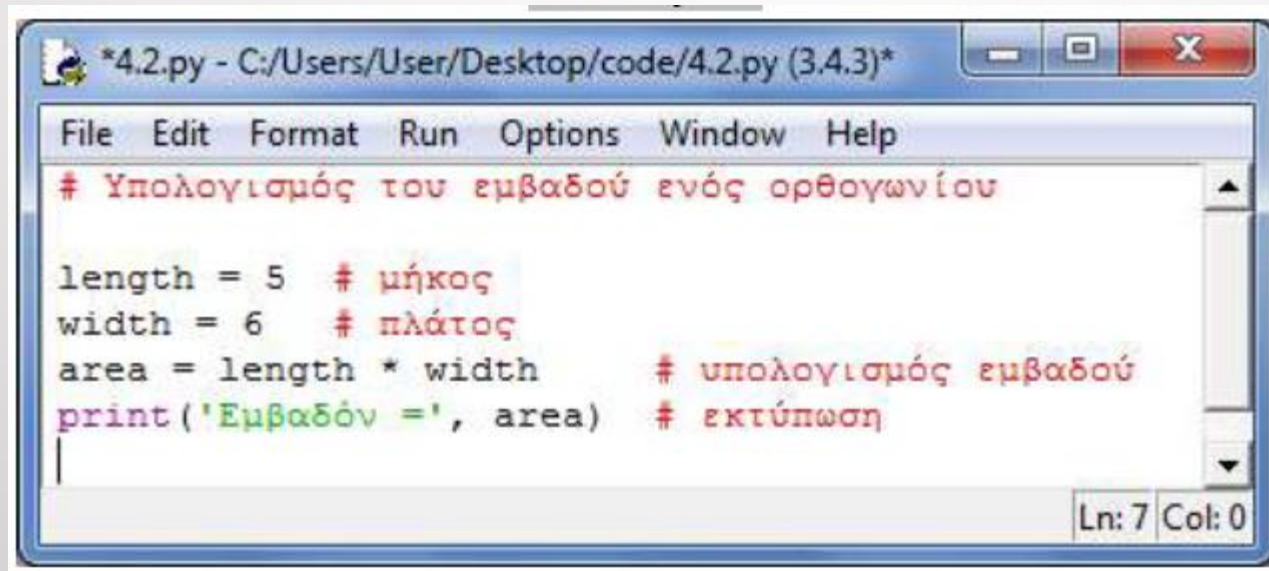
```
x = int(input('Δώστε έναν ακέραιο αριθμό:'))  
print(x, 'είναι ακέραιος αριθμός')  
print('Τέλος προγράμματος')
```

Σχόλια

- Καθώς γράφουμε όλο και μεγαλύτερα προγράμματα, μεγαλώνει και η δυσκολία να καταλαβαίνουμε αυτά που γράφουμε
- Χρειάζεται λοιπόν να προσθέτουμε **σχόλια (comments)** στα προγράμματα που γράφουμε
- Τα σχόλια στην Python αρχίζουν πάντα με τον χαρακτήρα #
- Οτιδήποτε ακολουθεί μετά το # αγνοείται από την Python μέχρι το τέλος της γραμμής

Σχόλια

- Παράδειγμα



```
*4.2.py - C:/Users/User/Desktop/code/4.2.py (3.4.3)*
File Edit Format Run Options Window Help
# Υπολογισμός του εμβαδού ενός ορθογωνίου

length = 5 # μήκος
width = 6 # πλάτος
area = length * width # υπολογισμός εμβαδού
print('Εμβαδόν =', area) # εκτύπωση
|
Ln: 7 Col: 0
```

Χαρακτήρες Διαφυγής

- **Χαρακτήρες Διαφυγής (Escape Characters)**
 - `\\` backslash
 - `'` μονά εισαγωγικά
 - `"` διπλά εισαγωγικά
 - `\n` νέα γραμμή
 - `\t` tab

Χαρακτήρες Διαφυγής

- Παράδειγμα

```
>>> print('\\Hello\\')
\\Hello\\
>>> print('It\\'s')
It's
>>> print('one\\ntwo\\nthree')
one
two
three
>>> print('one\\two')
one    two
>>>
```

Κάθε χαρακτήρας διαφυγής είναι ένας μονός χαρακτήρας, το σύμβολο \ δεν μετράει ως χαρακτήρας, απλά δηλώνει ότι πρόκειται για ειδικό χαρακτήρα

```
>>> len('It\\'s')
4
>>>
```

Ασκήσεις

1. Γράψτε ένα πρόγραμμα που θα διαβάζει τα στοιχεία ενός φοιτητή και την βαθμολογία 6 μαθημάτων του εξαμήνου και θα υπολογίζει το μέσο όρο του εξαμήνου
2. Γράψτε ένα πρόγραμμα που θα υπολογίζει το εμβαδό ορθογωνίου και τριγώνου
3. Γράψτε ένα πρόγραμμα που θα υπολογίζει την περίμετρο και το εμβαδό του κύκλου
4. Σχεδιάστε το διάγραμμα ροής της έκφρασης

$y = (\text{not } A) \text{ and } B \text{ or } C$

Υλοποιήστε σε γλώσσα Python το τμήμα του διαγράμματος ροής που αντιστοιχεί στην είσοδο και στην έξοδο των δεδομένων

Βιβλιογραφία

- Μανής, Γ., 2015. Εισαγωγή στον Προγραμματισμό με αρωγό τη γλώσσα Python. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: <http://hdl.handle.net/11419/2745>
- Αγγελιδάκης, Ν., 2015. Εισαγωγή στον προγραμματισμό με την Python, Ηράκλειο. [ηλεκτρ. βιβλ.] Διαθέσιμο στο: http://aggelid.mysch.gr/pythonbook/INTRODUCTION_TO_COMPUTER_PROGRAMMING_WITH_PYTHON.pdf