

Εργασία με λίστες

Μια μέρα, που λέτε, αποφάσισε ο Πύθωνας να καλέσει φίλους για φαγητό. Το ψυγείο και τα ντουλάπια ήταν άδεια, έτσι έπρεπε να πάει στην υπεραγορά να ψωνίσει. Επειδή είναι ξεχασιάρης, έπρεπε με κάποιο τρόπο να θυμηθεί πολλά και διαφορετικά πράγματα για να αγοράσει. Έτσι έφτιαξε μια λίστα!

Οι μεταβλητές, όπως τις είδαμε σε προηγούμενες σελίδες, μπορούν να κρατήσουν μια πληροφορία κάθε φορά. Για παράδειγμα, η μεταβλητή `myfood="banana"` κρατά την τιμή (περιεχόμενο) `"banana"` και τίποτα άλλο.

Θα μπορούσε, βέβαια, να χρησιμοποιήσει μια μεταβλητή που να περιέχει στα εισαγωγικά όλα τα πράγματα που θέλει π.χ.

```
myfood="banana, strawberry, raspberry".
```

Το πρόβλημα είναι πως θα μπερδευτεί: θα ψάχνει να βρει ΕΝΑ κουτί στην υπεραγορά που να τα έχει όλα αυτά μέσα!



Άρα, καλές οι μεταβλητές, όμως αν θέλουμε να αποθηκεύσουμε πολλές (και διαφορετικές) πληροφορίες, ώστε να βρίσκουμε την καθεμιά ξεχωριστά, θα χρησιμοποιήσουμε λίστες!

Μια λίστα έχει την ακόλουθη μορφή:

```
>>> mylist=["banana", "strawberry", "ice cream"]
```

Δίνουμε ένα όνομα στη λίστα, όπως θα το δίναμε και στη μεταβλητή (λίστα `"mylist"`) πιο πάνω. Στη συνέχεια, πληκτρολογούμε το `"="` ώστε να δώσουμε το περιεχόμενο της λίστας. Αν θα είναι κείμενο (string), γράφουμε το κάθε αντικείμενο ξεχωριστά με εισαγωγικά. Μια λίστα, σε αντίθεση με μια μεταβλητή, έχει το περιεχόμενο της μέσα στα σύμβολα `"["`.

Για να δούμε το περιεχόμενο της λίστας, πληκτρολογούμε την εντολή:

```
>>> print(my list)
```

```
['banana', 'cherry', 'strawberry']
```

Κάθε αντικείμενο σε μια λίστα έχει τη δική του σειρά: το πρώτο αντικείμενο βρίσκεται στη θέση 0, το δεύτερο στη θέση 1, το τρίτο στη θέση 2 κ.ο.κ. Σε επόμενο κεφάλαιο θα δούμε τη χρησιμότητα της λίστας και των θέσεων των αντικειμένων.



“Λίστες”, ‘Λίστες’, Λίστες!

Ας δούμε τις πιο κάτω λίστες:

```
mylist=['banana','icecream','strawberry']
```

```
mylist=["banana","icecream","strawberry"]
```

Και στις δύο περιπτώσεις, το αποτέλεσμα είναι το ίδιο - η λίστα mylist περιέχει τις ίδιες πληροφορίες σε μορφή string.

Σε περίπτωση που θέλουμε να χρησιμοποιήσουμε ακέραιους αριθμούς (integers) στη λίστα, θα πρέπει να έχει την εξής μορφή:

```
mynewlist=[10,14,23,44]
```

Για να δείξουμε το περιεχόμενο μιας λίστας, δε χρειάζεται

απαραίτητα η εντολή print(). Η print() είναι χρήσιμη όταν θέλουμε

να προβάσουμε και άλλες πληροφορίες μαζί με το

περιεχόμενο της λίστας.

Μπορούμε απλά να

πληκτρολογήσουμε το όνομα

της λίστας ώστε να



εμφανιστεί το περιεχόμενο της:

```
>>> mynewlist
      [10,14,23,44]
```

Αν έχουμε δύο λίστες, μπορούμε να ενώσουμε το περιεχόμενό τους. Για παράδειγμα:

```
>>> mylist1=["banana","watermelon"]
```

```
>>> mylist2=["apple","cherry"]
```

```
>>> mylist2=mylist2+mylist1
```

Στην τελευταία εντολή, βάζουμε στη λίστα mylist2 το περιεχόμενο που είχε (η mylist2) μαζί με το περιεχόμενο της mylist1. Αν πληκτρολογήσουμε mylist2, θα πάρουμε το ακόλουθο:

```
>>> mylist2
      ['apple', 'cherry', 'banana', 'watermelon']
```

Μπορούμε να προσθέσουμε περιεχόμενο σε μια λίστα με το σύμβολο "+" και τα αντικείμενα μέσα σε []. Για παράδειγμα:

```
mylist2=mylist2+["mango"]
```

Η λέξη mango θα προστεθεί στο τέλος της λίστας μας!



Αποφάσεις...

Οι πύθωνες είναι έξυπνα ζώα. Ένας πύθωνας ήθελε να αναβαθμίσει τον υπολογιστή του. Ήθελε να δει αν η μνήμη του ήταν μεγαλύτερη από 8GB. Σκέφτηκε "ΑΝ η μνήμη (του υπολογιστή) είναι μικρότερη από 8GB, θα την αναβαθμίσω". Πριν πάρει την απόφαση να αναβαθμίσει τη μνήμη του υπολογιστή του, πρώτα θα έπρεπε να ελέγξει κατά πόσο ήταν λιγότερη από 8GB (και, ξέρετε, με λιγότερα από τόσα δεν τρέχετε γρήγορα Windows 11 ή MacOS X ή νέες εκδόσεις του Linux).

Πώς θα το ελέγχαμε (σχεδόν) με ψευδοκώδικα:

```
"Αν η μνήμη (του υπολογιστή) < 8GB  
αναβάθμισε μνήμη"
```

Άρα, πρώτα θα ελέγξουμε αν η μνήμη μας είναι μικρότερη (το σύμβολο "<") από 8GB. Αν ισχύει αυτό, τότε και μόνο τότε θα αναβαθμίσουμε τη μνήμη του υπολογιστή!



Ας δούμε όμως σε κώδικα πώς δουλεύει αυτό:

```
>>> mymemory=8
```

Έχουμε δημιουργήσει μια μεταβλητή στην οποία δίνουμε την τιμή 8 (η μνήμη του υπολογιστή μας σε GB).

```
>>> if mymemory<8: print("Χρειάζεστε αναβάθμιση")
```

Η νέα εντολή "if" ζητά να ελέγξουμε αν ισχύει μια συνθήκη. Στην περίπτωση αυτή, ζητούμε να ελέγξει ΑΝ η τιμή της μεταβλητής είναι μικρότερη από 8. Σε περίπτωση που είναι μικρότερη από 8, τότε θα εμφανιστεί το μήνυμα Upgrade memory.

Στο παράδειγμα πιο πάνω, χρησιμοποιήσαμε μόνο μια συνθήκη (memory<8). Έτσι χρησιμοποιήσαμε μόνο μια γραμμή για τον κώδικα της συνθήκης.

Συνήθως στις συνθήκες ακολουθούμε την εξής δομή:

```
memory=8
if memory<8:
    print("Χρειάζεστε αναβάθμιση")
```

Το αποτέλεσμα βέβαια είναι το ίδιο, μιας και έχουμε και πάλι μία και μόνο συνθήκη να ελέγξουμε. Στη συνέχεια θα δούμε και άλλα παραδείγματα.

Μέχρι τώρα, ελέγχαμε έναν αριθμό και δίναμε μόνο μία απάντηση αν ισχυε η συνθήκη. Αν όμως δεν ισχύει, γιατί να μη δίνουμε κάποια πληροφορία; Εδώ θα χρησιμοποιήσουμε την εντολή "else".

Ας δώσουμε την εξής εντολή:

```
print("Χρειάζεστε αναβάθμιση") if mymemory<8 else
print("Δεν χρειάζεστε αναβάθμιση")
```

Στο πιο πάνω παράδειγμα, ακολουθήσαμε διαφορετική σύνταξη. Σε μία γραμμή ζητάμε να εμφανιστεί στην οθόνη το μήνυμα **Χρειάζεστε αναβάθμιση** **ΑΝ** η τιμή της mymemory είναι μικρότερη από 8, **ΔΙΑΦΟΡΕΤΙΚΑ** (else) να εμφανιστεί το μήνυμα **Δεν χρειάζεστε αναβάθμιση**.



Τις συνθήκες θα τις μελετήσουμε και σε επόμενο κεφάλαιο, σε μεγαλύτερο βάθος.

Είναι σημαντικό να κατανοήσουμε πως, τα παραδείγματα μας μέχρι τώρα αφορούσαν την εφαρμογή μιας εντολής κάθε φορά. Αργότερα θα γνωρίσουμε τον τρόπο δημιουργίας ενός σύνθετου προγράμματος που θα αποτελείται από πολλές γραμμές κώδικα.

Στις προηγούμενες σελίδες, γνωρίσαμε τη εντολή "if" και μελετήσαμε κάποια απλά παραδείγματα. Είναι σημαντικό να δούμε, στη συνέχεια, τον τρόπο με τον οποίο μπορούμε να χρησιμοποιήσουμε την εντολή αυτή για πιο σύνθετες συγκρίσεις.

Θα ξεκινήσουμε αρχικά με δύο μεταβλητές:

```
number1=10  
number2=20
```

Στις δύο αυτές μεταβλητές δώσαμε διαφορετικές τιμές.

Θέλουμε να συγκρίνουμε το περιεχόμενο των δύο μεταβλητών, ώστε να διαπιστώσουμε αν είναι το ίδιο.

```
if number1 == number2: print ("Equal")
```

Με την πιο πάνω εντολή, ελέγχουμε αν το περιεχόμενο τους είναι το ίδιο. Αυτό επιτυγχάνεται με το διπλό σύμβολο "=". Στο πιο πάνω παράδειγμα, η συνθήκη δεν ισχύει, έτσι δε θα εμφανιστεί κανένα μήνυμα.

Αν αυτό που μας ενδιαφέρει είναι να ελέγξουμε αν η τιμή τους είναι διαφορετική, τότε χρησιμοποιούμε την εντολή

```
if number1 != number2: print ("Equal")
```

Έλεγχοι που μπορούμε να κάνουμε:

- Μια μεταβλητή είναι μικρότερη από μια άλλη:
`number1 < number2`
- Μια μεταβλητή είναι μεγαλύτερη από μια άλλη:
`number1 > number2`
- Μια μεταβλητή είναι μικρότερη ή ίση από άλλη:
`number1 <= number2`
- Μια μεταβλητή είναι μεγαλύτερη ή ίση από άλλη:
`number1 >= number2`

Για συγκρίσεις, μπορούμε επίσης να χρησιμοποιήσουμε τα **And**, **Or** και **Not**.

```
if number1=100 and number2=100:  
print("Equal")
```

Στην πιο πάνω εντολή ελέγχουμε αν ισχύουν και οι δύο συνθήκες (και ο ένας αριθμός, και ο άλλος, ίσοι).

```
if number1=100 or number2=100: print("One hundred")
```

Στην πιο πάνω εντολή, ελέγχουμε αν έστω ο ένας από τους δύο ισούται με 100.