

Σύνθετοι Τύποι Δεδομένων: Δομές Δεδομένων



Οι μεταβλητές (variables) αποθηκεύουν μια τιμή κάθε φορά. Η τιμή αυτή μπορεί να είναι αριθμός, οποιοδήποτε σύμβολο, ή μια λογική τιμή. Για παράδειγμα:

```
>>> spam = 15
>>> spam = 'A'
>>> spam = True
```



Όταν όμως θέλουμε να αποθηκεύσουμε πολλές τιμές ως ένα ενιαίο σύνολο, τότε η Python δίνει τέσσερις βασικές δομές δεδομένων: [Συμβολοσειρές](#) (Strings), [Λίστες](#) (Lists), [Πλειάδες](#) (Tuples), [Λεξικά](#) (Dictionaries).

Οι Λίστες



Οι λίστες είναι συλλογές αντικειμένων, τα οποία μέσα στη λίστα έχουν συγκεκριμένη θέση. Η αρίθμηση των θέσεων ξεκινάει από το μηδέν, άρα δηλ. το πρώτο στοιχείο της λίστας βρίσκεται στη θέση 0.

Το χαρακτηριστικό σύμβολο που δείχνει ότι πρόκειται για λίστα είναι οι αγκύλες [και].

Ανοίξτε το IDLE και δοκιμάστε τα εξής:

```
>>> a = [10, 20, 30, 40]
>>> b = ['Hello', 'Καλημέρα']
>>> c = a + b
>>> c
>>> len(c)
```



Για να απομονώσουμε τμήματα μιας συμβολοσειράς, μετράμε την θέση κάθε γράμματος μέσα σε αυτήν, ξεκινώντας από το 0 (μηδέν). Πληκτρολογήστε τα εξής:

```
>>> a[0]           >>> b[1]           >>> c[6]
```



Μέσα στις αγκύλες μπορούμε να δώσουμε εύρος θέσεων. Δείτε πως ακριβώς λειτουργεί:

```
>>> a[0:2]         >>> c[0:len(c)]
```

Πως θα μπορούσατε να διαχωρίσετε τη λίστα a σε δύο λίστες, ώστε η πρώτη να περιέχει τα δύο πρώτα και η δεύτερη τα δύο τελευταία στοιχεία; Μια πιθανή λύση στην επόμενη σελίδα...

```
>>> a = [10, 20, 30, 40]
>>> protomiso = a[0:2]
>>> deyteromiso = a[2:len(a)]
```

Τι μπορούμε να κάνουμε με τις λίστες



Στο IDLE δημιουργήστε μία λίστα με όποια αντικείμενα θέλετε. Για παράδειγμα:

```
>>> mylist = ['Γαννης', 'Μαρία', 'Ελένη', 'Νίκος']
```

Για να προσθέσουμε στοιχεία στη λίστα, υπάρχει η εντολή `append()`. Δείτε πως ακριβώς δουλεύει:

```
>>> mylist.append('Πέτρος')
>>> mylist
>>> mylist.append(13)
>>> mylist
```

Τι παρατηρείτε;



Τα στοιχεία μίας λίστας μπορούν να αλλάξουν με απευθείας **αναθέση**, δηλ. με το `=`. Αυτό είναι κάτι που δεν γίνεται στις συμβολοσειρές.

Για να αλλάξουμε τον αριθμό 13 στην `mylist` πρέπει να γνωρίζουμε τη θέση του μέσα στη λίστα (και έστω ότι η `mylist` είναι `['Γαννης', 'Μαρία', 'Ελένη', 'Νίκος', 'Πέτρος', 13]`):

```
>>> mylist
>>> mylist[5] = 'Αλίκη'
>>> mylist
```

Ποιό περιμένετε να είναι το αποτέλεσμα της εντολής:

```
>>> mylist[20] = 'Σπύρος'
```

Πως το εξηγείτε;



Για να αφαιρέσουμε στοιχεία από μία λίστα, μπορούμε να χρησιμοποιήσουμε τις εντολές `pop()`, `remove()` και `del()`. Στο IDLE δοκιμάστε τα πιο κάτω παραδείγματα:

```
>>> a = [10, 20, 30, 40, 50]
>>> a.pop()
>>> a
```

12

```
>>> a.pop(1)
>>> a
```



Πως λειτουργεί η εντολή `pop()`;

Ας δούμε τώρα την εντολή `remove()`. Στο IDLE δοκιμάστε τα πιο κάτω παραδείγματα:



```
>>> a = [10, 20, 30, 40, 50]
>>> a.remove(20)
>>> a
>>> a.append(10)
>>> a
>>> a.remove(10)
```

Πως λειτουργεί η εντολή `remove()`;



Τι θα συμβεί αν δώσουμε:

```
>>> a.remove(500)
```

Η εντολή `del()` γράφεται διαφορετικά (έχει διαφορετική σύνταξη):



```
>>> a = [10, 20, 30, 40, 50]
>>> del(a[2])
>>> a
```

Πως λειτουργεί η εντολή `del()`;

Μερικές πιο ενδιαφέρουσες εντολές!



Στο IDLE δημιουργήστε μία κάπως μεγάλη συμβολοσειρά.

```
>>> s = 'Σήμερα είναι Κυριακή και δεν έχουμε σχολείο!'
```

Η εντολή `s.split()` χωρίζει τη συμβολοσειρά σε κομμάτια και με τα κομμάτια αυτά δημιουργεί μία λίστα. Δοκιμάστε τα εξής:

```
>>> list1 = s.split()
```

```
>>> list2 = s.split('α')
```

Οι δύο λίστες έχουν τα ίδια ή διαφορετικά στοιχεία; Πως το εξηγείτε;



Έστω ότι έχουμε ακόμη τις λίστες `list1` και `list2` που δημιουργήσατε πριν.

```
>>> list1
['Σήμερα', 'είναι', 'Κυριακή', 'και', 'δεν', 'έχουμε', 'σχολείο!']
>>> list2
['Σήμερ', ' είν', 'ι Κυρι', 'κή κ', 'ι δεν έχουμε σχολείο!']
```

Για να ελέγξουμε αν κάποιο στοιχείο υπάρχει μέσα στις λίστες, μπορούμε να χρησιμοποιήσουμε την εντολή `in`.

Στο IDLE δοκιμάστε τα εξής και εξηγήστε:

```
>>> 'και' in list1
```

```
>>> 'σχολείο' in list2
```

```
>>> 'σχολείο' in list1
```

Τι περιμένετε να δείτε ως αποτέλεσμα των πιο κάτω εντολών;

```
>>> list1 = 'Σήμερα είναι Κυριακή και δεν έχουμε σχολείο!'.split()
```

```
>>> x = 'και'
```

```
>>> y = 'λαλα'
```

```
>>> x in list1
```

```
>>> y in list1
```



Η εντολή for και οι λίστες



Στο IDLE δημιουργείτε μία συμβολοσειρά και μέσω αυτής μία λίστα.

```
>>> s = 'Star Wars – episode 2: attack of the clones'
>>> l = s.split()
>>> l
['Star', 'Wars', '-', 'episode', '2:', 'attack', 'of', 'the', 'clones']
```

Η εντολή for μπορεί να διαπεράσει τη λίστα και να ξεχωρίσει με τη σειρά ένα ένα τα αντικείμενα. Δοκιμάστε το εξής:

```
>>> for item in l:
    print(item)
```



Πολύ σημαντικό: ποιος είναι ο ρόλος της μεταβλητής `item` στην εντολή αυτή;

Στην παραπάνω συμβολοσειρά θέλουμε να μετρήσουμε πόσες λέξεις υπάρχουν.

Θυμηθείτε την εντολή `isalpha()`. Σε ένα νέο αρχείο, γράφουμε και συμπληρώνουμε τον πιο κάτω κώδικα.



```

starwars2.py - /
File Edit Format Run Options Window Help

s = 'Star Wars – episode 2: Attack of the clones'
l = ..... # δημιουργία λίστας από τη συμβολοσειρά

words = ..... # αρχική τιμή για τον μετρητή των λέξεων

for ..... in .....:
    if item.isalpha() == .....: # έλεγχος αν το στοιχείο είναι λέξη
        words = ..... + ..... # αύξηση μετρητή λέξεων

print('Η λίστα περιέχει ', words, ' λέξεις')
```

Στο IDLE δημιουργείτε τη λίστα `mylist` έτσι ώστε να περιέχει διάφορες λέξεις, κάποιες να ξεκινούν με κεφαλαίο γράμμα, κάποιες με μικρό, κάποιες να έχουν ανακατεμένα μικρά και κεφαλαία γράμματα, κλπ.

Δώστε την ακόλουθη εντολή και εξηγήστε πως ακριβώς λειτουργεί:

```
>>> mylist.sort()
```



Ασκήσεις



1. Τροποποιήστε τον κώδικα `starwars2.py` ώστε να μετράει πόσα αντικείμενα έχει η λίστα και πόσα από αυτά είναι λέξεις.

Δώστε μία εκδοχή με την εντολή `len()` και μία δεύτερη εκδοχή χωρίς την εντολή αυτή.



2. Γράψετε ένα νέο πρόγραμμα το οποίο θα διαβάζει από το πληκτρολόγιο 10 αριθμούς και θα τους αποθηκεύει σε λίστα. Στη συνέχεια, θα επεξεργάζεται τη λίστα και θα εμφανίζει:

(α) τον άθροισμα των αριθμών

(γ) τον μεγαλύτερο αριθμό

(β) τον μέσο όρο των αριθμών

(δ) τον μικρότερο αριθμό.



3. Στο πρόγραμμα της άσκησης 2 να προσθέσετε το εξής: να εμφανίζεται ο μεγαλύτερος αριθμός καθώς και πόσες φορές υπάρχει μέσα στη λίστα.



4. Να γράψετε ένα νέο πρόγραμμα το οποίο θα διαβάζει από το πληκτρολόγιο αριθμούς, μέχρι να δοθεί το 0. Οι θετικοί αριθμοί θα αποθηκεύονται στη λίστα `t` και οι αρνητικοί αριθμοί θα αποθηκεύονται στη λίστα `a`.