

Αντί εισαγωγής

Σήμερα θα δούμε τις **λίστες (lists)**. Όπως και τα **αλφαριθμητικά (strings)**, οι λίστες είναι **ακολουθίες (sequences)**. Όμως, ενώ τα αλφαριθμητικά είναι ακολουθίες από απλούς χαρακτήρες, οι λίστες είναι ακολουθίες από οτιδήποτε αντικείμενα: αριθμητικά, αλφαριθμητικά, ακόμη και από άλλες λίστες. Θυμίζουμε ότι στις ακολουθίες σημασία έχει η **σειρά** ή η **θέση** κάθε αντικειμένου, και όχι μόνο το περιεχόμενό του. Γι' αυτό το περιεχόμενο δύο ή περισσότερων αντικειμένων μέσα σε μια λίστα μπορεί να είναι το ίδιο (κάτι που δεν ισχύει στα σύνολα, όπως επίσης έχουμε πει). Κατά κανόνα στα προγράμματά μας τα αντικείμενα μιας λίστας είναι ομοιογενή, για παράδειγμα, είναι όλα αριθμοί, ή όλα αλφαριθμητικά, χωρίς αυτό να είναι απαραίτητο. Αρκετά με τη θεωρία, πάμε στην πράξη για μερικά **καινούρια** κόλπα της Python. Ξεκινάμε από το κέλυφος.



- >>>** 1. Φτιάξε μια άδεια λίστα με το όνομα numbers, και ζήτησε το περιεχόμενό της:

```
numbers = [] #ή numbers = list()
numbers
```

2. Τώρα άλλαξε το περιεχόμενό της:

```
numbers = [1, "two", 3.0, "IV"]
```

3. Η λίστα σου είναι ανομοιογενής. Λίγη χρησιμότητα θα μπορούσε να έχει. Λίγο παιχνίδι, πριν την εγκαταλείψεις. Δοκίμασε:

```
numbers + [5, "six"]
numbers * 3
```

4. Ξαναφτιάξε τη λίστα numbers:

```
numbers = [2, 4, 6, 8]
```

Τι θα γράψεις για να δεις το πρώτο στοιχείο της λίστας;

Τον δεύτερο;

Τον τελευταίο, με αρνητική αρίθμηση;

5. Προσάρτησε (πρόσθεσε στο τέλος της) στη numbers το στοιχείο 10, χρησιμοποιώντας τη μέθοδο append. Δες το αποτέλεσμα:

```
numbers.append(10)
```

6. Φτιάξε τώρα τη λίστα x με στοιχεία τους αριθμούς 12, 14, 16, 18, και 20;

.....

7. Προσάρτησε τη x στην numbers. Προσοχή! Δεν ζητείται να δείξεις το αποτέλεσμα της προσάρτησης, αλλά το αποτέλεσμα αυτό να ανανεώσει την numbers. Μπορείς να το κάνεις με δύο συμβολισμούς:

.....

8. Αντίστρεψε την numbers χρησιμοποιώντας τη μέθοδο reverse().

.....



Οι τελεστές + και * χαρακτηρίζονται από ισομορφισμό. Η πράξη που συμβολίζουν εξαρτάται από το είδος των δεδομένων στις οποίες την επιτελούν. Έτσι το «+» παριστάνει την **πρόσθεση** για αριθμητικά δεδομένα, αλλά τη **συνένωση** στα αλφαριθμητικά και τις λίστες.



Μια **μέθοδος (method)** είναι μια συνάρτηση που ανήκει σε συγκεκριμένα αντικείμενα, και μπορεί να χρησιμοποιηθεί μόνο για αυτά. Καλείται με τον τρόπο που είδες στο 5. Στο προηγούμενο φύλλο χρησιμοποιήσες 3 μεθόδους σε αλφαριθμητικά: την upper, την translate και την maketrans.



9. Όπως και στα αλφαριθμητικά, με τεμαχισμό (slicing), μπορούμε να πάρουμε τμήματα μιας λίστας. Τι θα γράψεις για να πάρεις από τη numbers, μια λίστα με:



- τα τρία πρώτα στοιχεία της numbers;
- τα τέσσερα τελευταία στοιχεία της numbers;
- Το 2ο, 5ο, 8ο, κλπ. στοιχεία της numbers;
- Τα στοιχεία στις άρτια αριθμημένες θέσεις, αλλά σε αντίστροφη σειρά, δηλαδή, το 20ο, 18ο, 16ο, έως το 2ο, στοιχεία;

10. Φτιάξε μια λίστα fruits με τα ονόματα των φρούτων "orange", "cherry", "apple", "banana", "orange" και "fig". Υπάρχει πρόβλημα που ένα στοιχείο είναι διπλό;

.....

11. Ταξινόμησε αλφαβητικά τη λίστα fruits χρησιμοποιώντας τη μέθοδο sort().

.....

12. Μπορείς να δεις τις μεθόδους ενός αντικειμένου με τη συνάρτηση dir. Γράψε:

```
dir(fruits) # ή dir(numbers) ή dir(list)
```

13. Για να δεις πληροφορίες για μια μέθοδο, έστω την pop, θα γράψεις:

```
help(fruits.pop)
```

14. Τι κατάλαβες για την pop. Δοκίμασέ την στην fruits χωρίς παράμετρο:

```
fruits.pop()
```

.....

.....

15. Δώσε την pop στη λίστα numbers με παράμετρο 3. Τι περιμένεις να συμβεί:

.....

.....

```
numbers.pop(3)
```

16. Η συνάρτηση len, όπως και στα αλφαριθμητικά, δηλώνει το πλήθος των στοιχείων μιας λίστας. Δώσε:

```
len(numbers)
```

17. Πώς θα εμφανίσεις μια λίστα σε ένα πρόγραμμά σου; Με την print():

```
for print(numbers)
```

18. Αν όμως δεν θέλεις τη συγκεκριμένη μορφοποίηση με τις αγκύλες και τα κόμματα; Θα θυμηθείς ότι η εντολή for είναι το ταίρι σε όλες τις ακολουθίες:

```
for fruit in fruits:
    print(fruit, end=" ")
```

19. Αρκετές φορές ο μετρητής μιας for πρέπει να συνδυαστεί με το δείκτη σε μια λίστα. Πώς θα αντικαταστήσεις τους αριθμούς στη λίστα numbers με τα τετράγωνά τους (στη θέση του 2 το 4, του 4 το 16, κ.ο.κ). Συμπλήρωσε κατάλληλα:

```
for i in range(....., .....):
    numbers.....
```



20. Ουφ! Κουράστηκες; Θα κλείσουμε με μια μικρή άσκηση στον συντάκτη. Οι αριθμοί Fibonacci είναι μια ακολουθία αριθμών που ορίζεται ως εξής. Ο πρώτος αριθμός Fibonacci είναι το μηδέν, ο δεύτερος το 1, και στη συνέχεια κάθε επόμενος, προκύπτει ως άθροισμα των δύο προηγούμενων: 1, 2, 3, 5,... Θα φτιάξεις το πρόγραμμα fib100.py που θα δημιουργεί τη λίστα fib με τους 100 πρώτους αριθμούς Fibonacci, και στη συνέχεια θα τους εμφανίζει στην οθόνη, έναν σε κάθε γραμμή:



```
fib=[..., ...]
for ... in .....:
    # Σε κάθε επανάληψη στη λίστα fib θα προσαρτάται ένα ακόμη στοιχείο
    .....
for .....:
    print(.....)
```



21. Τώρα θα γυρίσουμε στην κρεμάλα που έφτιαξες την περασμένη εβδομάδα. Θα δούμε μερικές απλές βελτιώσεις στον κώδικά της, και μερικές παραλλαγές και επεκτάσεις της.

ΜΠΡΑΒΟ!

- ✓ Ελπίζουμε να απόλαυσες την Python και αυτά τα φύλλα εργασίας όσο και εμείς!
- ✓ Συνέχισε να μελετάς και να πειραματίζεσαι.
- ✓ Στην τάξη του Edmodo, θα βρεις στη βιβλιογραφία, ένα μεγάλο πλήθος από πηγές, αξιόπιστες, δωρεάν, και ακόμη πιο απολαυστικές από τις 5 αυτές εβδομάδες.
- ✓ Εμείς θα είμαστε στη διάθεση σου για ηλεκτρονική επικοινωνία!

Συγχαρητήρια λοιπόν!

Να θυμάσαι ότι η γνώση είναι δύναμη!