

Ενότητα : Λίστες

Σχέδιο Δραστηριότητας: Εισαγωγή στις Λίστες

Εισαγωγή

Μια λίστα στην Python είναι ουσιαστικά ένας δυναμικός πίνακας. Δηλαδή ένας πίνακας όπως στην Pascal ή στην ΓΛΩΣΣΑ με τις εξής τρεις βασικές διαφορές:

1. Δεν χρειάζεται να τον δηλώσουμε κάπου
2. Δεν έχει σταθερό μέγεθος, μπορεί να αυξάνεται και να μειώνεται κατά την εκτέλεση του προγράμματος
3. Η αρίθμηση των δεικτών ξεκινάει από το 0 όπως συμβαίνει στη C και στην Java και παρέχεται η δυνατότητα αρνητικής δεικτοδότησης.

	0	1	2	3	4	5	6	7
	1	1	2	3	5	8	13	21
	-8	-7	-6	-5	-4	-3	-2	-1

Η Python παρέχει μεγάλη ποικιλία τελεστών και μεθόδων επεξεργασίας λιστών.

Δραστηριότητα 1

Ανοίξτε τον διερμηνευτή της Python, δώστε τις παρακάτω εντολές και παρατηρήστε τα αποτελέσματα. Να περιγράψετε τη λειτουργία των τελεστών **+**, **in**, των μεθόδων **pop** και **append** και των συναρτήσεων **len** και **range**.

<pre>>>> fib = [1, 1, 2, 3, 5, 8, 13, 21] >>> fib = fib + [34] >>> fib = [0] + fib >>> print fib >>> last = fib.pop() >>> print fib, last >>> fib.append(55) >>> print fib >>> 5 in fib</pre>	<pre>>>> len(fib) >>> fib.pop() ; len(fib) >>> range(10) >>> range(0, 10) >>> len(range(10)) >>> range(1, 10) >>> range(1, 10, 2) >>> range(10, 0, -2) >>> 100 not in range(1, 10)</pre>
--	---

Δραστηριότητα 2

Τα παρακάτω τμήματα κώδικα δημιουργούν μια λίστα με δυο τρόπους.

fib = [1, 1, 2, 3, 5, 8, 13, 21]	fib = [1, 1, 2, 3] + [5, 8, 13, 21]
----------------------------------	--------------------------------------

Να συμπληρώσετε το παρακάτω τμήμα κώδικα έτσι ώστε να δημιουργεί την ίδια λίστα και να ελέγξετε αν είναι σωστό στο IDLE. Στη συνέχεια τροποποιήστε το ώστε να κατασκευάζει μια λίστα με τους 100 πρώτους όρους της ακολουθίας Fibonaccι στην οποία κάθε όρος προκύπτει από το άθροισμα των δυο προηγούμενων.

```
fib = [1, 1 ]
for i in range(2, _____):
    fib[ i ] = fib[ _____ ] + fib[ _____ ]
```

Δραστηριότητα 3

Να γράψετε ένα πρόγραμμα το οποίο θα ζητάει από τον χρήστη έναν θετικό ακέραιο αριθμό N και στη συνέχεια θα εμφανίζει το άθροισμα όλων των αριθμών από το 1 έως και το N.

Δραστηριότητα 4

Το παρακάτω πρόγραμμα δημιουργεί μια λίστα με όλα τα θετικά πολλαπλάσια του 3 που είναι μικρότερα του 1000. Να το εκτελέσετε στο IDLE και να διορθώσετε τυχόν λάθη ώστε να βγάλει το σωστό αποτέλεσμα

```
list3 = [ ]
for i in range(1,1001, 3):
    list3 = list3 + i
```

Δραστηριότητα 5

Να γράψετε ένα πρόγραμμα το οποίο θα διαβάζει από το πληκτρολόγιο αριθμούς και θα τους αποθηκεύει σε μια λίστα. Η εισαγωγή των αριθμών θα σταματάει όταν δοθεί αρνητικός αριθμός. Στη συνέχεια το πρόγραμμα θα

1. Εμφανίζει πόσοι αριθμοί δόθηκαν
2. Εμφανίζει τους αριθμούς σε αντίστροφη σειρά από αυτή που δόθηκαν
3. Διαβάζει έναν αριθμό και θα ελέγχει αν έχει ήδη δοθεί

Δραστηριότητα 6

Να γράψετε ένα πρόγραμμα το οποίο δεδομένης μια λίστας ακέραιων αριθμών θα διαχωρίζει τους αριθμούς σε δυο νέες λίστες, μια για τους θετικούς και μια για τους αρνητικούς. Οι ομόσημοι αριθμοί πρέπει να παραμείνουν μεταξύ τους στη σειρά με την οποία δόθηκαν.

Εφαρμογή : Ιστόγραμμα Συχνοτήτων

Να αναπτύξετε ένα πρόγραμμα το οποίο αρχικά θα ζητάει από τον χρήστη έναν αριθμό N. Στη συνέχεια θα παράγει N τυχαίους αριθμούς στο διάστημα [1,6] και θα εμφανίζει τη συχνότητα των αριθμών με τη μορφή ιστογράμματος, όπως φαίνεται παρακάτω:

<pre>def frequency_histogram(N): numList = randomList(N) counter = frequency(numList) for i in range(1, 7): # για i από 1 μέχρι 6 histogram(i, counter[i])</pre>	<pre>>>> frequency_histogram(40) 1: * * * * * (10) 2: * * * * (5) 3: * * * (3) 4: * * * * * * * * * * (13) 5: * * * * * (7) 6: * * (2)</pre>
---	---

Ο σκελετός του προγράμματος σας δίνεται και εσείς αρκεί να υλοποιήσετε τις συναρτήσεις randomList, frequency, histogram ακολουθώντας τα παρακάτω βήματα:

Βήμα 1

Αρχικά θα πρέπει να υλοποιήσετε τη συνάρτηση **histogram**(number, times) η οποία θα δέχεται ένα αριθμό και το πλήθος των εμφανίσεών του και θα εμφανίζει το αντίστοιχο ιστόγραμμα. Όλα πρέπει να εμφανίζονται στην ίδια γραμμή.

Βήμα 2

Στη συνέχεια θα αναπτύξετε τη συνάρτηση **randomList**(number, times) η οποία θα δέχεται έναν αριθμό N, θα παράγει N ακέραιους αριθμούς με χρήση της *randint* της βιβλιοθήκης *random* στο διάστημα [1,6], θα τους αποθηκεύει σε μια λίστα και θα επιστρέφει τη λίστα ως τιμή της συνάρτησης.

Βήμα 3 (Η ώρα του υπολογισμού)

Τέλος σειρά έχει η συνάρτηση **frequency**(numList) η οποία θα δέχεται τη λίστα με τους αριθμούς και θα υπολογίζει τη συχνότητα κάθε αριθμού. Στη συνέχεια θα επιστρέφει τις συχνότητες με τη μορφή λίστας. Θα χρειαστούμε έναν μετρητή για κάθε αριθμό.

Επέκταση της εφαρμογής

Σκεφτείτε τι θα έπρεπε να αλλάξετε ώστε οι τυχαίοι αριθμοί να μην είναι στο διάστημα [1,6] αλλά στο διάστημα [1, N] με το N να δίνεται ως παράμετρο από τον χρήστη.