

# Ερωτηματολόγιο προγραμματισμού Python

**Απαντήσεις: Μάθημα 2 - Λίστες, λεξικά, πλειάδες και σύνολα**

**Revision:** 0.12

**Date:** 2009-07-12

**Author:** Savvas Radevic <[vicedar@gmail.com](mailto:vicedar@gmail.com)>

**Copyright:** © 2009 Savvas Radevic <[vicedar@gmail.com](mailto:vicedar@gmail.com)>, © 2009 python-gr group <[python-gr@googlegroups.com](mailto:python-gr@googlegroups.com)>

**Credits:** CREDITS.html

**License:** GNU FDL 1.3 <<http://www.gnu.org/licenses/fdl-1.3.txt>>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts. A copy of the license is included in the file entitled "COPYING-DOCS.txt".

**Source:** 02-lists-answers.tar.gz

**PDF:** 02-lists-answers.pdf

## Περιεχόμενα

<b>Γλωσσάρι λέξεων</b>	<b>1</b>
<b>Εισαγωγή</b>	<b>1</b>
<b>Ασκήσεις</b>	<b>1</b>
<b>Λίστες (και εν μέρει πλειάδες)</b>	<b>1</b>
<b>Λεξικά</b>	<b>7</b>
<b>Σύνολα</b>	<b>8</b>

## Γλωσσάρι λέξεων

Ελληνικά	English
Αντικείμενο / Στοιχείο / Στοιχείο δεδομένων	Item
Λίστα / Πίνακας	List
Λεξικό	Dictionary
Πλειάδα	Tuple
Αύξων αριθμός	Index number
Δυαδικοί τελεστές	Bitwise operators

## Εισαγωγή

Οποιοσδήποτε απορίες σχετικά με τις ασκήσεις / απαντήσεις, μπορείτε να τις εναποθέσετε στο θέμα συζήτησης: <http://forum.ubuntu-gr.org/viewtopic.php?f=55&t=5828>

## Ασκήσεις

## Λίστες (και εν μέρει πλειάδες)

1. Θέστε μια λίστα ως μεταβλητή με το όνομα `lista`.

Η μεταβλητή να περιέχει τα ακόλουθα στοιχεία:

- **'Μαρία'**
- **'Κώστας'**
- **'Αντώνης'**
- **1** (ως αριθμό)
- Την πράξη: **2 επί 3** (ως μαθηματική πράξη, όχι τον αριθμό 6 κατ' ευθείαν)

Απάντηση:

```
lista = ['Μαρία', 'Κώστας', 'Αντώνης', 1, 2*3]
```

Σε python2.6 το παραπάνω δουλεύει, αλλά δεν είναι unicode string:

```
$ python3
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> lista = ['Μαρία', 'Κώστας', 'Αντώνης', 1, 2*3]
>>> lista[0]
'Μαρία'
```

```
$ python2.6
Python 2.6.2 (release26-maint, Apr 19 2009, 01:58:18)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> lista = ['Μαρία', 'Κώστας', 'Αντώνης', 1, 2*3]
>>> lista[0]
'\xce\x9c\xce\xb1\xcf\x81\xce\xaf\xce\xb1'
```

Προσθέτουμε το γράμμα **u** πριν από κάθε **text string** για να το αντιπροσωπεύσουμε ως unicode text string:

```
>>> lista = [u'Μαρία', u'Κώστας', u'Αντώνης', 1, 2*3]
>>> lista[0]
u'\u039c\u03b1\u03c1\u03af\u03b1'
```

2. Πώς θα καλέσετε το στοιχείο δεδομένων με αύξων αριθμό 1 από την παραπάνω λίστα;

Απάντηση:

```
lista[1]
```

3. Αναφέρετε την εντολή ούτως ώστε να εμφανίσετε/εκτυπώσετε στην οθόνη το στοιχείο με αύξων αριθμό 2.

Απάντηση:

```
print(lista[2])
```

4. Ποια εντολή θα χρησιμοποιήσετε για να εκτυπώσετε σε μια γραμμή το **πρώτο** και **τρίτο** στοιχείο και την λέξη 'και' ενδιάμεσα, ούτως ώστε να εμφανιστεί ως

ακολουθως:

**Μαρία και Αντώνης**

Απάντηση:

```
print('{0} και {1}'.format(lista[0], lista[2]))
```

5. Χρησιμοποιώντας τον αύξων αριθμό των αντικειμένων/στοιχείων της λίστας `lista` (π.χ. `lista[1]`), δημιουργήστε μια καινούργια λίστα:

α. Ως μεταβλητή `lista3a` με τα **δύο πρώτα** στοιχεία της λίστας `lista` (ξεχωριστά στοιχεία, όχι υπο-λίστα).

Απάντηση:

```
lista3a = [lista[0], lista[1]]
```

β. Ως μεταβλητή `lista3b` με το **τελευταίο** στοιχείο της λίστας `lista` (**ΧΩΡΙΣ** να χρησιμοποιήσετε `lista[4]!`)

**Σημείωση:** Μπορώ να σκεφτώ 2 τρόπους χωρίς το `lista[4]`, εσείς; :)

Απάντηση:

```
lista3b = lista[len(lista)-1]  
lista3b = lista[-1]
```

γ. Ως μεταβλητή `lista3c` με τα ακόλουθα στοιχεία:

- Ως υπο-λίστα το **πρώτο και δεύτερο** στοιχείο της λίστας `lista`. Χρησιμοποιήστε το `lista[x:y]` όπου `x` και `y` να είναι οι αύξων αριθμοί που χρειάζονται.
- Ως υπο-λίστα την λίστα `lista3a`.

Απάντηση:

```
lista3c = [lista[0:2], lista3a]
```

6. Χρησιμοποιώντας αντικείμενα με τον αύξων αριθμό τους (index number) της λίστας `lista`, δημιουργήστε μια καινούργια λίστα:

α. Ως μεταβλητή `lista4a` με το **προτελευταίο** στοιχείο της λίστας `lista` (βοήθεια: αρνητικοί αριθμοί).

Απάντηση:

```
lista4a = lista[-2]
```

β. Ως μεταβλητή `lista4b` με τα στοιχεία **από το πρώτο μέχρι το προτελευταίο** (βοήθεια: ερώτηση 5γ).

Απάντηση:

```
lista4b = lista[0:-1]
```

7. Φτιάξτε καινούργια λίστα-αντίγραφο της λίστας `lista`. Ονομάστε την `lista7`.

Απάντηση:

```
lista7 = lista
```

α. Αναφέρετε την εντολή με την οποία μπορείτε να αλλάξετε/αντικαταστήσετε το τέταρτο στοιχείο της λίστας `lista7` (δλδ το στοιχείο με τον αριθμό **1**).

Απάντηση:

```
lista7[3] = 2
```

β. Μετατρέψτε **και αντικαταστήστε** το τέταρτο στοιχείο της λίστας `lista7`, προσθέστε στην τιμή του **+5** (αναφέροντας την μαθηματική πράξη στην εντολή, όχι τον αριθμό 6 κατ' ευθείαν).

Απάντηση:

```
lista7[3] = lista7[3] + 5
```

8. Ποια εντολή θα χρησιμοποιήσετε για να δείτε το "μέγεθος" (τον αριθμό των αντικειμένων) της λίστας `lista`;

Απάντηση:

```
# Σημειώστε πως επιστρέφει ΑΡΙΘΜΟ, όχι text string!  
len(lista)
```

9. Αναφέρετε τρόπο με τον οποίο μπορείτε να καθαρίσετε μια λίστα (δλδ. να σβήσετε όλα τα στοιχεία).

Απάντηση:

```
lista7 = []
```

Επίσης:

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)  
[GCC 4.3.3] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> del(lista7[:])  
>>> lista7  
[]
```

10. Φτιάξτε μια λίστα ως μεταβλητή `lista10`:

```
lista10 = [1, 2, 3, 4, 5]
```

Ακολουθως:

α. προσθέστε στη λίστα `lista10` μια υπο-λίστα που να περιέχει τα ακόλουθα στοιχεία:

- **100**
- **200**

Απάντηση:

```
lista10 = [1, 2, 3, 4, 5, [100, 200]]
```

β. καλέστε τον αριθμό **100** ως στοιχείο της λίστας `lista10`.

Απάντηση:

```
lista10[5][0]
```

γ. Φτιάξτε μια πλειάδα-αντίγραφο της λίστας lista10. Ονομάστε την πλειάδα pleiada10. (βοήθεια: tuple())

Απάντηση:

```
pleiada10 = tuple(lista10)
```

11. Έχουμε την ακόλουθη λίστα:

```
#           0           1           2           3           4
lista11 = ['μήλα', 'αχλάδια', 'αγγούρια', 'μπανάνες', 'κιτρόμηλα']
```

Σημείωση: Για κάθε υπο-ερώτηση που ακολουθεί χρησιμοποιείτε τη λίστα που δημιουργήσατε από το **προηγούμενο** μέρος της άσκησης, **όχι την αρχική lista11!** Δηλαδή παίρνετε τη λίστα-αποτέλεσμα που φτιάξατε στο μέρος (α) και την χρησιμοποιείτε στο μέρος (β).

α. χρησιμοποιώντας το lista11.insert(), προσθέστε **META** τις 'μπανάνες' ένα **NEO** στοιχείο: **'ντομάτες'**

Απάντηση:

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> lista11 = ['μήλα', 'αχλάδια', 'αγγούρια', 'μπανάνες', 'κιτρόμηλα']
>>> lista11.insert(4, 'ντομάτες')
>>> lista11
['μήλα', 'αχλάδια', 'αγγούρια', 'μπανάνες', 'ντομάτες', 'κιτρόμηλα']
```

β. χρησιμοποιώντας το lista11[0:1] = ... και το lista11[0], προσθέστε **PRIN** το πρώτο στοιχείο ('μήλα') ένα **NEO** στοιχείο: **'πατάτες'**

Απάντηση:

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> lista11[-2::]
['ντομάτες', 'κιτρόμηλα']
>>> lista11[0:1] = ['πατάτες', lista11[0]]
>>> lista11
['πατάτες', 'μήλα', 'αχλάδια', 'αγγούρια', 'μπανάνες', 'ντομάτες', 'κιτρόμηλα']
```

γ. χρησιμοποιώντας το lista11[-2::] = ... και το lista11[-1], προσθέστε στο ΤΕΛΟΣ (δλδ. σαν τελευταίο στοιχείο) ένα **NEO** στοιχείο: **'κεράσια'**

Απάντηση:

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> lista11[-2::] = ['ντομάτες', lista11[-1], 'κεράσια']
>>> lista11
['πατάτες', 'μήλα', 'αχλάδια', 'αγγούρια', 'μπανάνες', 'ντομάτες', 'κιτρόμηλα', 'κεράσια']
```

δ. χρησιμοποιώντας το lista11.pop(), αφαιρέστε τα 'κεράσια'.

Απάντηση:

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> listall.pop()
'κεράσια'
>>> listall
['πατάτες', 'μήλα', 'αχλάδια', 'αγγούρια', 'μπανάνες', 'κιτρόμηλα']
```

ε. αφαιρέστε τα στοιχεία που είναι φρούτα, με μια ή περισσότερες εντολές (βοήθεια: [Η ντομάτα είναι φρούτο](#))

Απάντηση:

**Σημείωση:** Τελικά, έχοντας ως βάση τη βοτανολογία (μιας και την ίδια βάση πήραμε προηγουμένως για τη ντομάτα), και το αγγούρι [είναι φρούτο](#). Αν και δεν ήταν πρόθεση μου να παραπλανήσω, ευχαριστώ τον [kon\\_nos που το ανέφερε](#) για τις απαντήσεις του! :)

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> listall
['πατάτες', 'μήλα', 'αχλάδια', 'αγγούρια', 'μπανάνες', 'ντομάτες', 'κιτρόμηλα', 'κεράσια']
>>> listall.remove('μήλα')
>>> listall
['πατάτες', 'αχλάδια', 'αγγούρια', 'μπανάνες', 'ντομάτες', 'κιτρόμηλα', 'κεράσια']
>>> listall[4::]
['ντομάτες', 'κιτρόμηλα', 'κεράσια']
>>> listall[4::] = []
>>> listall
['πατάτες', 'αχλάδια', 'αγγούρια', 'μπανάνες']
>>> listall.pop()
'μπανάνες'
>>> listall
['πατάτες', 'αχλάδια', 'αγγούρια']
>>> listall = [listall[0]]
>>> listall
['πατάτες']
```

**Σημείωση:** Για να μην έχουμε παρεξηγήσεις, δε χρειάζονται όλοι τα βήματα εδώ, το τελευταίο αρκεί! απλά είπα να σας δείξω τί επιλογές έχετε όταν θέλετε να διαγράψετε ένα αντικείμενο.

12. Έχουμε την ακόλουθη αριθμητική σειρά:

**0, 13, 4, 26, 8, 39, ...**

```
lista12 = [0, 13, 4, 26, 8, 39]
```

α. βρέστε ποιος είναι ο επόμενος αριθμός.

**Σημείωση:** Είναι ερώτηση λογικής! Δε χρειάζεται κώδικας / αλγόριθμος, απλώς λίγη σκέψη.. :)

Απάντηση: **12** - Η σειρά περιέχει δύο σειρές: 0,4,8,12 (+4) και 13-26-39 (+13)

β. προσθέστε τον αριθμό από το μέρος (α) χρησιμοποιώντας το `lista12.append()`.

Απάντηση:

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> lista12 = [0, 13, 4, 26, 8, 39]
>>> lista12.append(12)
```

```
>>> lista12
[0, 13, 4, 26, 8, 39, 12]
```

## Λεξικά

1. Φτιάξτε ένα λεξικό:

```
leksiko = {
    'Όνομα'      : 'Savvas',
    'Επίθετο'   : 'Radevic',
    'Πόλη'      : 'Λευκωσία',
    1           : 'Αριθμός!'
}
```

Απάντηση:

```
leksiko = {
    'Όνομα'      : 'Savvas',
    'Επίθετο'   : 'Radevic',
    'Πόλη'      : 'Λευκωσία',
    1           : 'Αριθμός!'
}
```

**Σημείωση:** Από ό,τι βλέπετε τα κενά που χρησιμοποιείτε μεταξύ των κλειδιών(keys)/μεταβλητών και των τιμών (values) δεν παίζουν ρόλο. Ακόμη ένας λόγος να **ωραιοποιήσετε** τον κώδικα σας κάνοντας τον πιο σαφή και πιο εύκολο στο διάβασμα!

α. Εκτυπώστε με την εντολή `print()` ένα οποιοδήποτε κείμενο με τα παραπάνω στοιχεία χρησιμοποιώντας `leksiko['Όνομα']` για το όνομα, `leksiko['Επίθετο']` για το επίθετο και `leksiko['Πόλη']` για την πόλη.

Απάντηση:

```
print("Η αίτησή σας έχει καταχωρηθεί, σημειώνοντας πως είναι από την πόλη (2) υπό το όνομα (0) (1). Θα επικοινωνήσουμε μαζί σας το συντομότερο δυνατό.".format(leksiko['Όνομα'], leksiko['Επίθετο'], leksiko['Πόλη']))
```

β. Φτιάξτε λεξικό-αντίγραφο του `leksiko`, θέστε το ως `leksiko2` και αφαιρέστε τα αντικείμενα **'Πόλη'** και **1** από το `leksiko2`.

Απάντηση:

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> leksiko2 = leksiko
>>> leksiko2
{1: 'Αριθμός!', 'Επίθετο': 'Radevic', 'Όνομα': 'Savvas', 'Πόλη': 'Λευκωσία'}

>>> del(leksiko2['Πόλη'], leksiko2[1])
>>> leksiko2
{'Επίθετο': 'Radevic', 'Όνομα': 'Savvas'}
```

γ. Προσθέστε το αντικείμενο **'Ζώδιο'** με την τιμή **'Ζυγός'**.

Απάντηση:

```
leksiko['Ζώδιο'] = 'Ζυγός'
```

δ. Πώς θα ελέγξετε αν το **'Ζώδιο'** είναι αντικείμενο του λεξικού `leksiko`;

Απάντηση:

```
>>> 'Ζώδιο' in leksiko
True
```

**Σημείωση:** Το "True" συμβολίζει "Σωστό" (Ναι), ενώ το "False" συμβολίζει το "Λάθος" (Όχι).

ε. Αντικαταστήστε την τιμή του αντικειμένου '**Πόλη**', από 'Λευκωσία' σε 'Βελιγράδι', στο λεξικό leksiko.

Απάντηση:

```
leksiko['Πόλη'] = 'Βελιγράδι'
```

στ. Δημιουργήστε μια λίστα lista\_leksiko με τα στοιχεία '**Όνομα**' και '**Επίθετο**'.

Απάντηση:

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> lista_leksiko = [leksiko['Όνομα'], leksiko['Επίθετο']]
>>> lista_leksiko
['Savvas', 'Radevic']
```

## Σύνολα

1. Έχουμε την ακόλουθη λίστα:

```
#           0           1           2           3           4
lista_set = ['μήλα', 'αχλάδια', 'αγγούρια', 'μπανάνες', 'κιτρόμηλα']
```

Δημιουργήστε ένα σύνολο fruit με την εντολή **set()** από την lista\_set.

Απάντηση:

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> lista_set = ['μήλα', 'αχλάδια', 'αγγούρια', 'μπανάνες', 'κιτρόμηλα']
>>> fruit = set(lista_set)
>>> fruit
{'κιτρόμηλα', 'μπανάνες', 'μήλα', 'αχλάδια', 'αγγούρια'}
```

### Σημειώσεις:

- Προσέξτε ότι τα σύνολα στην python3 είναι εσωκλεισμένα σε {}.
- **MHN** το μπλέξετε με τα λεξικά, τα οποία έχουν μεταβλητές-κλειδιά και τιμές!!
- Στην python2.6 συμβολίζονται εσωκλεισμένα σε set().

```
Python 2.6.2 (release26-maint, Apr 19 2009, 01:58:18)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> lista_set = ['μήλα', 'αχλάδια', 'αγγούρια', 'μπανάνες', 'κιτρόμηλα']
>>> fruit = set(lista_set)
>>> fruit
set(['\u03b1\u03b2\u03b3\u03b4\u03b5\u03b6\u03b7\u03b8\u03b9', '\u03b1\u03b2\u03b3\u03b4\u03b5\u03b6\u03b7\u03b8\u03b9', '\u03b1\u03b2\u03b3\u03b4\u03b5\u03b6\u03b7\u03b8\u03b9', '\u03b1\u03b2\u03b3\u03b4\u03b5\u03b6\u03b7\u03b8\u03b9', '\u03b1\u03b2\u03b3\u03b4\u03b5\u03b6\u03b7\u03b8\u03b9', '\u03b1\u03b2\u03b3\u03b4\u03b5\u03b6\u03b7\u03b8\u03b9', '\u03b1\u03b2\u03b3\u03b4\u03b5\u03b6\u03b7\u03b8\u03b9', '\u03b1\u03b2\u03b3\u03b4\u03b5\u03b6\u03b7\u03b8\u03b9', '\u03b1\u03b2\u03b3\u03b4\u03b5\u03b6\u03b7\u03b8\u03b9', '\u03b1\u03b2\u03b3\u03b4\u03b5\u03b6\u03b7\u03b8\u03b9'])
```

2. Δημιουργήστε τα σύνολα numbers\_negative (αριθμοί από -10 μέχρι 0) και numbers\_positive (αριθμοί από 0 μέχρι 10):

```
numbers_negative = set(range(-10,1))
numbers_positive = set(range(0,11))
```

α. Δημιουργήστε το σύνολο `numbers_superset`, το οποίο είναι **υπερσύνολο** των συνόλων `numbers_negative` και `numbers_positive`. (βοήθεια: χρησιμοποιήστε ένα από τους **bitwise operators**, που σημαίνει ότι το στοιχείο ανήκει είτε στο πρώτο είτε στο δεύτερο σύνολο)

Απάντηση:

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> numbers_negative = set(range(-10,1))
>>> numbers_positive = set(range(0,11))
>>> numbers_superset = numbers_positive | numbers_negative
>>> numbers_superset
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, -2, -10, -9, -8, -7, -6, -5, -4, -3, -1}
```

**Σημείωση:** Όπως βλέπετε, η σειρά δεν είναι ταξινομημένη. Τα σύνολα δεν είναι για ταξινόμηση, αλλά κυρίως για έλεγχο υπερσυνόλων / υποσυνόλων. Στα σύνολα το πώς είναι τα δεδομένα και με ποια ακολουθία εμφανίζονται είναι αμελητέο, αφού ενδιαφερόμαστε για τα δεδομένα που περιέχουν τα σύνολα.

β. Δημιουργήστε το σύνολο `numbers_common`, στο οποίο βρίσκονται αριθμοί που ανήκουν **και στα δύο σύνολα**, `numbers_negative` και `numbers_positive`. (βοήθεια: χρησιμοποιήστε ένα από τους **bitwise operators**)

Απάντηση:

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> numbers_common = numbers_positive & numbers_negative
>>> numbers_common
{0}
```

γ. Δημιουργήστε ένα σύνολο `numbers_subset`:

Πώς θα ελέγξετε αν το `numbers_subset` είναι υποσύνολο του `numbers_positive`;

Απάντηση:

```
>>> numbers_subset = set(range(1,5))
>>> numbers_subset.issubset(numbers_positive)
True
```