

Λίστες στην Python

Κεφάλαιο 8

Python για Όλους

www.py4e.com



Προγραμματισμός

- **Αλγόριθμος**
 - Ένα σύνολο κανόνων ή βημάτων που στόχο έχουν την επίλυση ενός προβλήματος
- **Δομή Δεδομένων**
 - Ένας λεπτομερής τρόπος οργάνωσης δεδομένων σε έναν υπολογιστή

<https://en.wikipedia.org/wiki/Algorithm>

https://en.wikipedia.org/wiki/Data_structure

Τι δεν είναι μια «Συλλογή»;

Οι περισσότερες από τις μεταβλητές μας έχουν μία τιμή - όταν βάζουμε μια νέα τιμή στη μεταβλητή, η παλιά τιμή αντικαθίσταται

```
$ python
>>> x = 2
>>> x = 4
>>> print(x)
4
```

Μια Λίστα είναι ένα Είδος Συλλογής



- Μια **συλλογή** μας επιτρέπει να βάλουμε πολλές τιμές σε μία «**μεταβλητή**»
- Μια **συλλογή** είναι ωραία γιατί μπορούμε να μεταφέρουμε **πολλές τιμές** σε ένα βολικό πακέτο.

φίλοι = ['Ιωάννης', 'Δέσποινα', 'Ελένη']

αποσκευή = ['κάλτσες', 'πουκάμισο', 'άρωμα']

Λίστα Σταθερών

- Οι **λίστες** σταθερών περιβάλλονται από τετράγωνες αγκύλες και τα στοιχεία τους χωρίζονται με κόμμα
- Ένα στοιχείο **λίστας** μπορεί να είναι οποιοδήποτε αντικείμενο Python - ακόμη και μια **άλλη λίστα**
- Μια **λίστα** μπορεί να είναι κενή

```
>>> print([1, 24, 76])
[1, 24, 76]
>>> print(['red', 'yellow',
'blue'])
['red', 'yellow', 'blue']
>>> print(['red', 24, 98.6])
['red', 24, 98.6]
>>> print([1, [5, 6], 7])
[1, [5, 6], 7]
>>> print([])
[]
```

Χρησιμοποιήσαμε Ήδη Λίστες!

```
for i in [5, 4, 3, 2, 1] :  
    print(i)  
print('Εκτόξευση!')
```

5

4

3

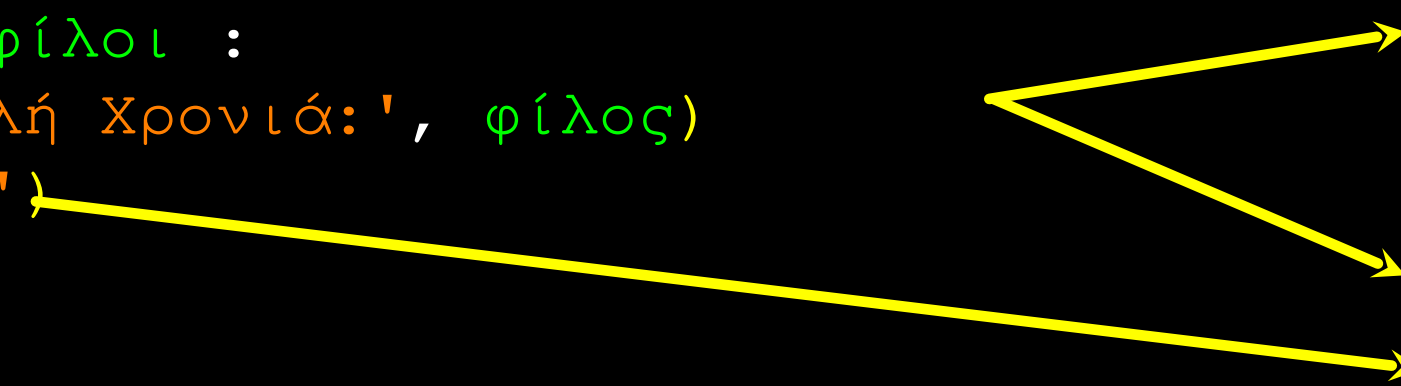
2

1

Εκτόξευση!

Λίστες και Καθορισμένοι Βρόχοι – Οι Καλύτεροι Φίλοι

```
φίλοι = ['Ιωσήφ', 'Κατερίνα', 'Σπύρο']  
for φίλος in φίλοι :  
    print('Καλή Χρονιά:', φίλος)  
print('Τέλος!')
```



Καλή Χρονιά: Ιωσήφ
Καλή Χρονιά: Κατερίνα
Καλή Χρονιά: Σπύρο
Τέλος!

```
z = ['Ιωσήφ', 'Κατερίνα', 'Σπύρο']  
for x in z:  
    print('Καλή Χρονιά:', x)  
print('Τέλος!')
```



Κοιτάζοντας Μέσα σε Λίστες

Ακριβώς όπως οι συμβολοσειρές, μπορούμε να πάρουμε οποιοδήποτε στοιχείο μιας λίστας χρησιμοποιώντας έναν δείκτη, που γράφεται μέσα σε **αγκύλες**

Ιωσήφ	Κατερίνα	Σπύρο
0	1	2

```
>>> φίλοι = ['Ιωσήφ', 'Κατερίνα', 'Σπύρο']  
>>> print(φίλοι[1])  
Κατερίνα  
>>>
```

Οι Λίστες είναι Μεταβλητές

- Οι συμβολοσειρές είναι «**αμετάβλητες**» - δεν μπορούμε να αλλάξουμε το περιεχόμενο μιας συμβολοσειράς - πρέπει να κάνουμε μια **νέα συμβολοσειρά** για να κάνουμε οποιαδήποτε αλλαγή
- Οι λίστες είναι «**μεταβλητές**» - μπορούμε να **αλλάξουμε** ένα στοιχείο μιας λίστας χρησιμοποιώντας τον τελεστή **δείκτη**

```
>>> φρούτο = 'Banana'
>>> φρούτο[0] = 'b'
Traceback
TypeError: 'str' object does not
support item assignment
>>> x = φρούτο.lower()
>>> print(x)
banana
>>> lotto = [2, 14, 26, 41, 63]
>>> print(lotto)
[2, 14, 26, 41, 63]
>>> lotto[2] = 28
>>> print(lotto)
[2, 14, 28, 41, 63]
```

Τι Μήκος έχει μια Λίστα;

- Η συνάρτηση `len()` παίρνει μια **λίστα** ως παράμετρο και επιστρέφει τον αριθμό των **στοιχείων** στη **λίστα**
- Στην πραγματικότητα το `len()` μας δίνει τον αριθμό των στοιχείων οποιουδήποτε συνόλου ή ακολουθίας (π.χ. μια συμβολοσειρά ...)

```
>>> greet = 'Hello Bob'
>>> print(len(greet))
9
>>> x = [ 1, 2, 'joe', 99]
>>> print(len(x))
4
>>>
```

Χρησιμοποιώντας τη Συνάρτηση

range

- Η συνάρτηση `range` σε συνδυασμό με τη `list` επιστρέφει μια λίστα αριθμών που κυμαίνονται από μηδέν έως έναν μικρότερο από την παράμετρο
- Μπορούμε να δημιουργήσουμε έναν βρόχο δεικτών χρησιμοποιώντας `for` και έναν ακέραιο επαναλήπτη

```
>>> print(list(range(4)))
[0, 1, 2, 3]
>>> φίλοι = ['Ιωσήφ', 'Κατερίνα', 'Σπύρο']
>>> print(len(φίλοι))
3
>>> print(list(range(len(φίλοι))))
[0, 1, 2]
>>>
```

Μια Ιστορία Δύο Βρόχων...

```
φίλοι = ['Ιωσήφ', 'Κατερίνα', 'Σπύρο']
```

```
for φίλος in φίλοι :  
    print('Καλή Χρονιά:', φίλος)
```

```
for i in range(len(φίλοι)) :  
    φίλος = φίλοι[i]  
    print('Καλή Χρονιά:', φίλος)
```

```
>>> φίλοι = ['Ιωσήφ', 'Κατερίνα', 'Σπύρο']  
>>> print(len(φίλοι))  
3  
>>> print(list(range(len(φίλοι))))  
[0, 1, 2]  
>>>
```

Καλή Χρονιά: Ιωσήφ

Καλή Χρονιά: Κατερίνα

Καλή Χρονιά: Σπύρο

Συνένωση Λιστών με Χρήση +

Μπορούμε να δημιουργήσουμε μια νέα λίστα προσθέτοντας δύο υπάρχουσες λίστες μαζί

```
>>> a = [1, 2, 3]
>>> b = [4, 5, 6]
>>> c = a + b
>>> print(c)
[1, 2, 3, 4, 5, 6]
>>> print(a)
[1, 2, 3]
```

Οι Λίστες Μπορούν να Τεμαχιστούν με Χρήση του :

```
>>> t = [9, 41, 12, 3, 74, 15]
>>> t[1:3]
[41, 12]
>>> t[:4]
[9, 41, 12, 3]
>>> t[3:]
[3, 74, 15]
>>> t[:]
[9, 41, 12, 3, 74, 15]
```

Θυμηθείτε: Ακριβώς όπως στις συμβολοσειρές, ο δεύτερος αριθμός είναι «μέχρι, αλλά δεν περιλαμβάνετε»

Μέθοδοι Λιστών

```
>>> x = list()
>>> type(x)
<type 'list'>
>>> dir(x)
['append', 'count', 'extend', 'index', 'insert',
'pop', 'remove', 'reverse', 'sort']
>>>
```

<http://docs.python.org/tutorial/datastructures.html>

Δημιουργία Λίστας από την Αρχή

- Μπορούμε να δημιουργήσουμε μια κενή **λίστα** και στη συνέχεια να προσθέσουμε στοιχεία χρησιμοποιώντας τη μέθοδο **append**
- Η **λίστα** παραμένει σε σειρά και τα νέα στοιχεία **προστίθενται** στο τέλος της **λίστας**

```
>>> stuff = list()
>>> stuff.append('book')
>>> stuff.append(99)
>>> print(stuff)
['book', 99]
>>> stuff.append('cookie')
>>> print(stuff)
['book', 99, 'cookie']
```

Υπάρχει Κάτι στη Λίστα;

- Η Python παρέχει δύο **τελεστές** που σας επιτρέπουν να ελέγξετε εάν ένα στοιχείο βρίσκεται σε μια λίστα
- Αυτοί είναι λογικοί τελεστές που επιστρέφουν **True** ή **False**
- Δεν τροποποιούν τη λίστα

```
>>> some = [1, 9, 21, 10, 16]
>>> 9 in some
True
>>> 15 in some
False
>>> 20 not in some
True
>>>
```

Οι λίστες είναι σε Σειρά

- Μια **λίστα** μπορεί να περιέχει πολλά στοιχεία και τα διατηρεί στη σειρά μέχρι να κάνουμε κάτι για να αλλάξουμε τη σειρά τους
- Μια **λίστα** μπορεί να **ταξινομηθεί** (δηλαδή, να αλλάξει τη σειρά της)
- Η μέθοδος **sort** (σε αντίθεση με τις συμβολοσειρές) σημαίνει «**ταξινόμησε τον εαυτό σου**»

```
>>> φίλοι = ['Κατερίνα', 'Σπύρο', 'Ιωσήφ']
>>> φίλοι.sort()
>>> print(φίλοι)
['Ιωσήφ', 'Κατερίνα', 'Σπύρο']
>>> print(φίλοι[1])
Κατερίνα
>>>
```

Ενσωματωμένες Συναρτήσεις και Λίστες

- Υπάρχουν πλήθος **συναρτήσεων** ενσωματωμένες στην **Python** που λαμβάνουν **λίστες** ως παραμέτρους
- Θυμάστε τους βρόχους που χτίσαμε; Αυτά είναι πολύ πιο απλά.

```
>>> nums = [3, 41, 12, 9, 74, 15]
>>> print(len(nums))
6
>>> print(max(nums))
74
>>> print(min(nums))
3
>>> print(sum(nums))
154
>>> print(sum(nums)/len(nums))
25.6
```

```
άθροισμα = 0
πλήθος = 0
while True :
    inp = input('Δώστε έναν αριθμό: ')
    if inp == 'τέλος' : break
    τιμή = float(inp)
    άθροισμα = άθροισμα + τιμή
    πλήθος = πλήθος + 1
```

```
μέσος_όρος = άθροισμα / πλήθος
print('Μέσος Όρος:', μέσος_όρος)
```

Δώστε έναν αριθμό: 3

Δώστε έναν αριθμό: 9

Δώστε έναν αριθμό: 5

Δώστε έναν αριθμό: τέλος

Μέσος Όρος: 5.666666666666667

```
numlist = list()
while True :
    inp = input('Δώστε έναν αριθμό: ')
    if inp == 'τέλος' : break
    τιμή = float(inp)
    numlist.append(τιμή)
```

```
μέσος_όρος = sum(numlist) / len(numlist)
print('Μέσος Όρος:', μέσος_όρος)
```

Καλύτερες Φίλες: Συμβολοσειρές και Λίστες

```
>>> abc = 'With three words'  
>>> stuff = abc.split()  
>>> print(stuff)  
['With', 'three', 'words']  
>>> print(len(stuff))  
3  
>>> print(stuff[0])  
With
```

```
>>> print(stuff)  
['With', 'three', 'words']  
>>> for w in stuff :  
...     print(w)  
...  
With  
Three  
Words  
>>>
```

Το **Split** σπάει μια συμβολοσειρά σε μέρη και παράγει μια λίστα συμβολοσειρών. Αυτά τα θεωρούμε λέξεις. Μπορούμε να έχουμε **πρόσβαση** σε μια συγκεκριμένη λέξη ή να **διατρέξουμε** όλες τις λέξεις.

```
>>> γραμμή = 'A lot of spaces'
>>> etc = γραμμή.split()
>>> print(etc)
['A', 'lot', 'of', 'spaces']
>>>
>>> γραμμή = 'πρώτο;δεύτερο;τρίτο'
>>> αντικ = γραμμή.split()
>>> print(αντικ)
['πρώτο;δεύτερο;τρίτο']
>>> print(len(αντικ))
1
>>> αντικ = γραμμή.split(';')
>>> print(αντικ)
['πρώτο ', 'δεύτερο ', 'τρίτο ']
>>> print(len(αντικ))
3
>>>
```

- Όταν δεν καθορίζετε έναν οριοθέτη, πολλαπλά κενά αντιμετωπίζονται σαν ένας οριοθέτης
- Μπορείτε να καθορίσετε ποιον χαρακτήρα οριοθέτησης θα χρησιμοποιήσετε στο διαχωρισμό

From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008

```
fhand = open('mbox-short.txt')
for γραμμή in fhand:
    γραμμή = γραμμή.rstrip()
    if not γραμμή.startswith('From ') : continue
    λέξεις = γραμμή.split()
    print(λέξεις[2])
```

Sat
Fri
Fri
Fri
...

```
>>> γραμμή = 'From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008'
>>> λέξεις = γραμμή.split()
>>> print(λέξεις)
['From', 'stephen.marquard@uct.ac.za', 'Sat', 'Jan', '5', '09:14:16', '2008']
>>>
```

Το Μοτίβο Διπλού Διαχωρισμού

Μερικές φορές χωρίζουμε μια γραμμή με έναν τρόπο και στη συνέχεια πιάνουμε ένα από τα κομμάτια της γραμμής και το χωρίζουμε ξανά

From **stephen.marquard@uct.ac.za** Sat Jan 5 09:14:16 2008

```
λέξεις = γραμμή.split()  
email = λέξεις[1]
```

Το Μοτίβο Διπλού Διαχωρισμού

From **stephen.marquard@uct.ac.za** Sat Jan 5 09:14:16 2008

```
λέξεις = γραμμή.split()
```

```
email = λέξεις[1]
```

```
stephen.marquard@uct.ac.za
```

Το Μοτίβο Διπλού Διαχωρισμού

From **stephen.marquard@uct.ac.za** Sat Jan 5 09:14:16 2008

```
λέξεις = γραμμή.split()
```

```
email = λέξεις[1]
```

```
τμήματα = email.split('@')
```

```
stephen.marquard@uct.ac.za
```

```
['stephen.marquard', 'uct.ac.za']
```

Το Μοτίβο Διπλού Διαχωρισμού

From **stephen.marquard@uct.ac.za** Sat Jan 5 09:14:16 2008

```
λέξεις = γραμμή.split()
email = λέξεις[1]
τμήματα = email.split('@')
print(τμήματα[1])
```

stephen.marquard@uct.ac.za
['stephen.marquard', 'uct.ac.za']
'uct.ac.za'

Σύνοψη

- Έννοια της συλλογής
- Λίστες και καθορισμένοι βρόχοι
- Ευρετηρίαση και αναζήτηση
- Μεταβλητότητα λίστας
- Συναρτήσεις: `len`, `min`, `max`, `sum`
- Τεμαχισμός λιστών
- Μέθοδοι λιστών: `append`, `remove`
- Ταξινόμηση λιστών
- Τεμαχισμός συμβολοσειρών σε λίστες λέξεων
- Χρήση `split` για ανάλυση συμβολοσειρών



Ευχαριστίες / Συνεισφορές



Αυτές οι διαφάνειες είναι Πνευματική ιδιοκτησία 2010- Charles R. Severance (www.dr-chuck.com) του University of Michigan School of Information και είναι διαθέσιμες υπό την άδεια Creative Commons Attribution 4.0. Παρακαλώ να διατηρήσετε αυτήν την τελευταία διαφάνεια σε όλα τα αντίγραφα του εγγράφου για να συμμορφωθείτε με τις απαιτήσεις απόδοσης της άδειας. Εάν κάνετε κάποια αλλαγή, μη διστάσετε να προσθέσετε το όνομα και τον οργανισμό σας στη λίστα των συντελεστών αυτής της σελίδας καθώς αναδημοσιεύετε το υλικό.

Συνέχεια...

Αρχική ανάπτυξη : Charles Severance, University of Michigan School of Information

Απόδοση στα Ελληνικά: Κιουρτίδου Δ. Κωνσταντία

... Εισαγάγετε νέους Μεταφραστές και άτομα που έχουν συνεισφέρει εδώ