Λεξικά της Python

Κεφάλαιο 9

Python για Όλους

www.py4e.com

Τι είναι μια Συλλογή;

- Μια συλλογή είναι ωραία γιατί μπορούμε να βάλουμε περισσότερες από μία τιμές σε αυτήν και να τις μεταφέρουμε σε ένα βολικό πακέτο
- Έχουμε μια δέσμη τιμών σε μία μόνο «μεταβλητή»
- Αυτό το κάνουμε έχοντας περισσότερες από μία θέσεις «μέσα» στη μεταβλητή
- Έχουμε τρόπους εύρεσης των διαφορετικών θέσεων στη μεταβλητή

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 Ιστορία Δύο Συλλογών…

- Λίστα

  - Μια γραμμική συλλογή τιμών που παραμένουν σε σειρά

- Λεξικό

  - Ένας «σάκος» τιμών, η καθεμιά με τη δική του ετικέτα

Λεξικά

http://en.wikipedia.org/wiki/Associative_array

Λεξικά

- Τα λεξικά είναι η πιο ισχυρή συλλογή δεδομένων της Python
- Τα λεξικά μας επιτρέπουν να κάνουμε γρήγορες λειτουργίες, που μοιάζουν με βάση δεδομένων, στην Python
- Τα λεξικά έχουν διαφορετικά ονόματα σε διαφορετικές γλώσσες

  - Πίνακες Συσχέτισης - Perl / PHP

  - Ιδιότητες ή Χάρτης ή HashMap - Java

  - Property Bag - C# / .Net

Λεξικά

- Οι λίστες ευρετηριάζουν τις καταχωρίσεις τους με βάση τη θέση στη λίστα
- Τα λεξικά είναι σαν τις τσάντες – καμία σειρά
- Έτσι ευρετηριάζουμε τα πράγματα που βάζουμε στο λεξικό με μια "ετικέτα αναζήτησης"

```
>>> τσάντα = dict()
>>> τσάντα['χρήματα'] = 12
>>> τσάντα['καραμέλα'] = 3
>>> τσάντα['χαρτομάντηλα'] = 75
>>> print(τσάντα)
{'χρήματα': 12, 'χαρτομάντηλα': 75, 'καραμέλα': 3}
>>> print(τσάντα['καραμέλα'])
3
>>> τσάντα['καραμέλα'] = τσάντα['καραμέλα'] + 2
>>> print(τσάντα)
{'χρήματα ': 12, 'χαρτομάντηλα ': 75, 'καραμέλα': 5}
```

Συγκρίνοντας Λίστες και Λεξικά

Τα λεξικά είναι σαν λίστες, εκτός από το ότι χρησιμοποιούν κλειδιά αντί αριθμών για την αναζήτηση τιμών

```
>>> lst = list()
>>> lst.append(21)
>>> lst.append(183)
>>> print(lst)
[21, 183]
>>> lst[0] = 23
>>> print(lst)
[23, 183]
```

```
>>> ddd = dict()
>>> ddd['ηλικία'] = 21
>>> ddd['μάθημα'] = 182
>>> print(ddd)
{'ηλικία': 21, 'μάθημα': 182}
>>> ddd['ηλικία'] = 23
>>> print(ddd)
{'ηλικία': 23, 'μάθημα': 182}
```

```
>>> lst = list()
>>> lst.append(21)
>>> lst.append(183)
>>> print(lst)
[21, 183]
>>> lst[0] = 23
>>> print(lst)
[23, 183]
```

Λίστα

Κλειδί	Τιμή
[0]	21
[1]	183

lst

```
>>> ddd = dict()
>>> ddd['ηλικία'] = 21
>>> ddd['μάθημα'] = 182
>>> print(ddd)
{'ηλικία': 21, 'μάθημα': 182}
>>> ddd['ηλικία'] = 23
>>> print(ddd)
{'ηλικία': 23, 'μάθημα': 182}
```

Λεξικό

Κλειδί	Τιμή
['ηλικία']	21
['μάθημα']	182

ddd

Περιεχόμενο Λεξικών (σταθερές)

- Τα περιεχόμενα των λεξικών περικλείονται από άγκιστρα και έχουν μια λίστα με ζεύγη κλειδιού: τιμής

- Μπορείτε να δημιουργήσετε ένα άδειο λεξικό χρησιμοποιώντας άδεια άγκιστρα

```
>>> jjj = {'Κάρολος' : 1 , 'Μαρία' : 42, 'Ιωνας': 100}
>>> print(jjj)
{'Κάρολος' : 1 , 'Μαρία' : 42, 'Ιωνας': 100}
>>> ooo = { }
>>> print(ooo)
{}
>>>
```

Το πιο Συχνά Εμφανιζόμενο Όνομα;

Το πιο Συχνά Εμφανιζόμενο Όνομα;

Μάρκος

Ζαχαρίας

Κάρολος

Ζαχαρίας

Γεωργία

Μάρκος

Ζαχαρίας

Κάρολος

Γεωργία

Ζαχαρίας

Κάρολος

Μάρκος

Ζαχαρίας

Το πιο Συχνά Εμφανιζόμενο Όνομα;

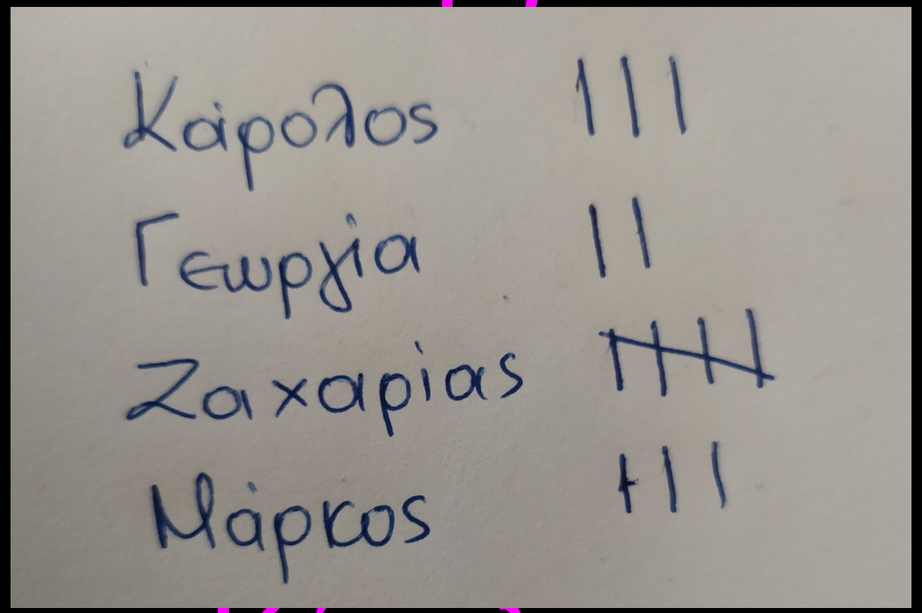
Μάρκος

Ζαχαρίας

Κάρολος

Ζαχαρίας

Γεωργία



Κάρολος

Γεωργία

Ζαχαρίας

Κάρολος

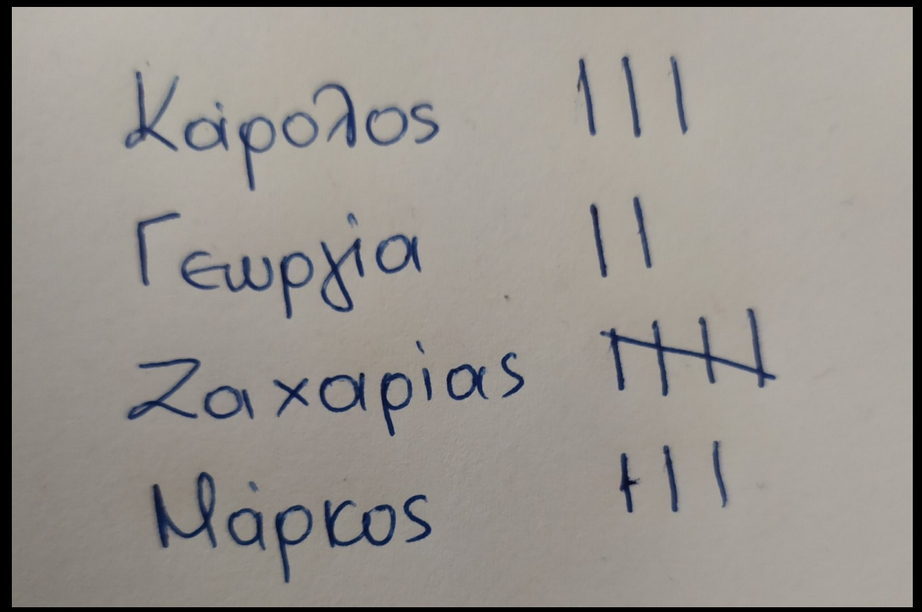
Μάρκος

Ζαχαρίας

Πολλοί Μετρητές με Ένα Λεξικό

Μια συνήθης χρήση των λεξικών είναι η μέτρηση του πόσο συχνά «βλέπουμε» κάτι

```
>>> ccc = dict()
>>> ccc['Κάρολος'] = 1
>>> ccc['Γεωργία'] = 1
>>> print(ccc)
{'Κάρολος': 1, 'Γεωργία': 1}
>>> ccc['Γεωργία'] = ccc['Γεωργία'] + 1
>>> print(ccc)
{'Κάρολος': 1, 'Γεωργία': 2}
```



Traceback Λεξικών

- Είναι λάθος να αναφερθείτε σε ένα κλειδί που δεν υπάρχει στο λεξικό

- Μπορούμε να χρησιμοποιήσουμε τον τελεστή in για να ελέγξουμε αν ένα κλειδί υπάρχει στο λεξικό

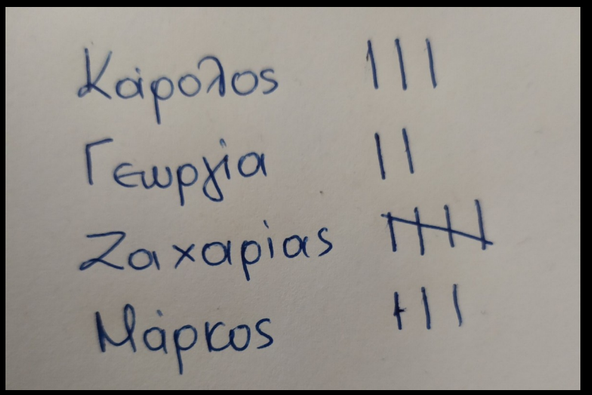
```
>>> ccc = dict()
>>> print(ccc['Κάρολος'])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'Κάρολος'
>>> 'Κάρολος' in ccc
False
```

Όταν Βλέπουμε Ένα Νέο Όνομα

Όταν συναντάμε ένα νέο όνομα, πρέπει να προσθέσουμε μια νέα καταχώριση στο λεξικό και αν αυτή είναι η δεύτερη ή μεταγενέστερη φορά που έχουμε δει το όνομα, απλά προσθέτουμε ένα στο πλήθος στο λεξικό με αυτό το όνομα

```
πλήθη = dict()
ονόματα = ['Κάρολος', 'Γεωργία', 'Κάρολος', 'Ζαχαρίας', 'Γεωργία']
for όνομα in ονόματα :
    if όνομα not in πλήθη:
        πλήθη[name] = 1
    else :
        πλήθη[name] = πλήθη[name] + 1
print(πλήθη)
```

{'Κάρολος': 2, 'Γεωργία': 2 , 'Ζαχαρίας': 1}



Η Μέθοδος get για τα Λεξικά

Το μοτίβο του ελέγχου για να διαπιστωθεί εάν ένα κλειδί βρίσκεται ήδη σε ένα λεξικό και υποθέτοντας μια προεπιλεγμένη τιμή, εάν το κλει`δί δεν είναι εκεί, είναι τόσο κοινό που υπάρχει μια μέθοδος που ονομάζεται get() που το κάνει αυτό για εμάς

```
if όνομα in πλήθη:
    x = πλήθη[όνομα]
else :
    x = 0
```

```
x = πλήθη.get(όνομα, 0)
```

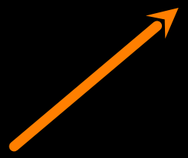
Προεπιλεγμένη τιμή εάν το κλειδί δεν υπάρχει (και όχι Traceback).

{'Κάρολος': 2, 'Γεωργία': 2 , 'Ζαχαρίας': 1}

Απλοποιημένη Μέτρηση με get()

Μπορούμε να χρησιμοποιήσουμε το get() και να δώσουμε μια προεπιλεγμένη τιμή μηδέν όταν το κλειδί δεν είναι ακόμη στο λεξικό - και στη συνέχεια μόνο να προσθέτουμε ένα

```
πλήθη = dict()
ονόματα = ['Κάρολος', 'Γεωργία', 'Κάρολος', 'Ζαχαρίας', 'Γεωργία']
for όνομα in ονόματα :
    πλήθη[name] = πλήθη.get(όνομα, 0) + 1
print(πλήθη)
```



Προεπιλεγμένο

{'Κάρολος': 2, 'Γεωργία': 2 , 'Ζαχαρίας': 1}

Απλοποιημένη Μέτρηση με get()

```
πλήθη = dict()
ονόματα = ['Κάρολος', 'Γεωργία', 'Κάρολος',
'Ζαχαρίας', 'Γεωργία']
for όνομα in ονόματα :
    πλήθη[όνομα] = πλήθη.get(όνομα, 0) + 1
print(πλήθη)
```

http://www.youtube.com/watch?v=EHJ9uYx5L58

Μετρώντας Λέξεις σε Κείμενο

Η συγγραφή προγραμμάτων (ή προγραμματισμός) είναι μια πολύ δημιουργική και ανταποδοτική δραστηριότητα. Μπορείτε να γράψετε προγράμματα για πολλούς λόγους που κυμαίνονται από το να βγάλετε τα προς το ζην έως την επίλυση ενός δύσκολου προβλήματος ανάλυσης δεδομένων, ή τη διασκέδασή σας, ή τη βοήθεια σε κάποιον άλλο για να λύσει ένα πρόβλημα. Αυτό το βιβλίο υποθέτει ότι όλοι πρέπει να γνωρίζουν πώς να προγραμματίζουν και ότι μόλις μάθετε πώς να προγραμματίζετε, θα καταλάβετε τι θέλετε να κάνετε με τις νέες δεξιότητές σας.

Περιτριγυριζόμαστε, στην καθημερινότητά μας, από υπολογιστές που κυμαίνονται από φορητούς υπολογιστές έως κινητά τηλέφωνα. Μπορούμε να σκεφτούμε αυτούς τους υπολογιστές ως τους «προσωπικούς βοηθούς» μας που μπορούν να φροντίσουν πολλά πράγματα για λογαριασμό μας. Οι σημερινοί υπολογιστές είναι ουσιαστικά φτιαγμένο για να μας κάνουν συνεχώς την ερώτηση «Τι θα θέλατε να κάνω στη συνέχεια;»

Οι υπολογιστές μας είναι γρήγοροι και έχουν τεράστια ποσότητα μνήμης και θα μπορούσαν να μας βοηθήσουν πολύ αν γνωρίζαμε τη γλώσσα τους για να εξηγήσουμε στον υπολογιστή μας τι θα θέλαμε να κάνει στη συνέχεια. Αν γνωρίζαμε αυτή τη γλώσσα θα μπορούσαμε να πούμε στον υπολογιστή να κάνει εργασίες για λογαριασμό μας, που ήταν επαναλαμβανόμενες. Είναι ενδιαφέρον ότι τα πράγματα που μπορούν να κάνουν οι υπολογιστές είναι συχνά τα πράγματα που εμείς οι άνθρωποι τα θεωρούμε βαρετά και ενοχλούν.

Μοτίβο Μέτρησης

```
πλήθη = dict()
print('Δώστε μια γραμμή κειμένου:')
γραμμή = input('')

λέξεις = γραμμή.split()

print('Λέξεις:', λέξεις)

print('Μέτρηση...')
for λέξη in λέξεις:
    πλήθη[λέξη] = πλήθη.get(λέξη,0) + 1
print('Πλήθη', πλήθη)
```

Το γενικό μοτίβο για την καταμέτρηση των λέξεων σε μια γραμμή κειμένου είναι η διαίρεση της γραμμής σε λέξεις, μετά η περιήγηση στις λέξεις και η χρήση ενός λεξικού για την παρακολούθηση του πλήθους κάθε λέξης ανεξάρτητα.

```
python wordcount.py
Δώστε μια γραμμή κειμένου:
the clown ran after the car and the car ran into the tent
and the tent fell down on the clown and the car

Λέξεις: ['the', 'clown', 'ran', 'after', 'the', 'car',
'and', 'the', 'car', 'ran', 'into', 'the', 'tent', 'and',
'the', 'tent', 'fell', 'down', 'on', 'the', 'clown',
'and', 'the', 'car']
Μέτρηση…

Πλήθη {'and': 3, 'on': 1, 'ran': 2, 'car': 3, 'into': 1,
'after': 1, 'clown': 2, 'down': 1, 'fell': 1, 'the': 7,
'tent': 2}
```

http://www.flickr.com/photos/71502646@N00/2526007974/

```
πλήθη = dict()
γραμμή = input('Δώστε μια γραμμή κειμένου:')
λέξεις = γραμμή.split()

print('Λέξεις:', λέξεις)
print('Μέτρηση...’)

for λέξη in λέξεις:
     πλήθη[λέξη] = πλήθη.get(λέξη,0) + 1
print('Πλήθη', πλήθη)
```

```
python wordcount.py
'Δώστε μια γραμμή κειμένου:'
the clown ran after the car and the car ran
into the tent and the tent fell down on the
clown and the car

Λέξεις: ['the', 'clown', 'ran', 'after', 'the', 'car',
'and', 'the', 'car', 'ran', 'into', 'the', 'tent', 'and',
'the', 'tent', 'fell', 'down', 'on', 'the', 'clown',
'and', 'the', 'car’]
Μέτρηση...

Πλήθη {'and': 3, 'on': 1, 'ran': 2, 'car': 3,
'into': 1, 'after': 1, 'clown': 2, 'down': 1, 'fell':
1, 'the': 7, 'tent': 2}
```

Καθορισμένοι Βρόχοι και Λεξικά

Παρόλο που τα λεξικά δεν αποθηκεύονται με τη σειρά, μπορούμε να γράψουμε έναν βρόχο for, που περνάει από όλες τις καταχωρήσεις ενός λεξικού - στην πραγματικότητα περνάει από όλα τα κλειδιά του λεξικού και αναζητά τις τιμές

```
>>> πλήθη = {'κάρολος' : 1 , 'φανή' : 42, 'ίων': 10}
>>> for κλειδί in πλήθη:
...     print(κλειδί, πλήθη[κλειδί])
...
κάρολος 1
φανή 42
ίων 10
>>>
```

Ανάκτηση Λίστας Κλειδιών και Τιμών

Μπορείτε να λάβετε μια λίστα με κλειδιά, τιμές ή στοιχεία (και τα δύο) από ένα λεξικό

```
>>> jjj = {'κάρολος' : 1 , 'φανή' : 42, 'ίων': 10}
>>> print(list(jjj))
['κάρολος', 'φανή', 'ίων']
>>> print(list( jjj.keys() ))
['κάρολος', 'φανή', 'ίων']
>>> print(list(jjj.values()))
[1, 42, 10]
>>> print(list( jjj.items() ))
[('κάρολος', 1), ('φανή', 42), ('ίων', 10)]
>>>
```

Αυτή είναι μια «πλειάδα»; - προσεχώς...

Μπόνους: Δύο Μεταβλητές Επανάληψης!

- Διατρέχουμε στα ζεύγη κλειδιών-τιμών σε ένα λεξικό χρησιμοποιώντας * δύο* μεταβλητές επανάληψης

- Σε κάθε επανάληψη, η πρώτη μεταβλητή είναι το κλειδί και η δεύτερη μεταβλητή είναι η αντίστοιχη τιμή για το κλειδί

```
jjj = {'κάρολος' : 1 , 'φανή' : 42, 'ίων': 10}
for aaa,bbb in jjj.items() :
    print(aaa, bbb)
```

```
κάρολος 1
φανή 42
ίων 10
```

aaa	bbb
[κάρολος]	1
[φανή]	42
[ίων]	10

```
όνομα = input('Δώστε αρχείο:')
handle = open(όνομα)

πλήθη = dict()
for γραμμή in handle:
    λέξεις = line.split()
    for λέξη in λέξεις:
        πλήθη[λέξη] = πλήθη.get(λέξη,0) + 1

μαχπλήθος = None
μαχλέξη = None
for λέξη,πλήθος in πλήθη.items():
    if μαχπλήθος is None or πλήθος > μαχπλήθος:
        μαχλέξη = λέξη
        μαχπλήθος = πλήθος

print(μαχλέξη, μαχπλήθος)
```

```
python words.py
Enter file: words.txt
to 16
```

```
python words.py
Enter file: clown.txt
the 7
```

Χρησιμοποιώντας δύο ενφωλευμένους βρόχους

Σύνοψη

- Τι είναι μία συλλογή;
- Λίστες εναντίων Λεξικών
- Σταθερές Λεξικών
- Η πιο συχνή λέξη
- Χρήση της μεθόδου get()
- Κατακερματισμός, έλλειψη σειράς
- Βρόχοι λεξικών
- Κρυφοκοιτώντας τις πλειάδες
- Ταξινόμηση λεξικών

Ευχαριστίες / Συνεισφορές

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

Συνέχεια…