

Σημειώσεις

[Προηγούμενο](#)[Επόμενο](#)

Λεξικά

Τα λεξικά είναι μία ακόμα δομή δεδομένων την οποία παρέχει η Python. Μπορεί να την φανταστεί κανείς σε αναλογία με τα γνωστά λεξικά, όπου γνωρίζοντας μία λέξη-κλειδί (key) οδηγούμαστε σε μία πληροφορία. Το λεξικό που θα δούμε εδώ αποτελείται από μία σειρά λέξεις-κλειδιά (keys) το καθένα από τα οποία συνδέεται με μία αντίστοιχη πληροφορία (object). Κάθε εγγραφή στο λεξικό είναι της μορφής

```
key: object
```

Ένα λεξικό ορίζεται με τη βοήθεια αγκίστρων {}, ανάμεσα στα οποία βάζουμε όλες τις εγγραφές. Μπορούμε να βρούμε μία εγγραφή δίνοντας την αντίστοιχη λέξη-κλειδί σε ορθογώνιες παρενθέσεις (όπως στις λίστες και συμβολοσειρές).

```
age = {"Mihalis": 48, "Manolis": 50, "Nikos": 12}
age["Nikos"] += 2
print(age["Nikos"])
print(age)
```

Μπορούμε να προσθέσουμε εγγραφές στο λεξικό.

```
>>> D = {"Mihalis": 48, "Manolis": 50, "Nikos": 12}
>>> name = input("Give name: ")
Give name: Yannis
>>> if name not in age: D[name] = 55
>>> print(D)
{'Mihalis': 48, 'Manolis': 50, 'Nikos': 12, 'Yannis': 55}
```

Μπορούμε να σβήσουμε εγγραφές από το λεξικό.

```
>>> name = input("Give name: ")
Give name: Nikos
>>> if name in age: del age[name]
>>> print(D)
{'Mihalis': 48, 'Manolis': 50, 'Yannis': 55}
```

Μπορούμε να διατρέξουμε ένα λεξικό με το `for`.

```
>>> D = {'Mihalis': 48, 'Manolis': 50, 'Nikos': 12, 'Yannis': 55}
>>> for key in D:
...     print("The age of",key,"is",D[key])
...
The age of Mihalis is 48
The age of Manolis is 50
The age of Nikos is 12
The age of Yannis is 55
```

Παρατηρήσεις.

- Δεν επιτρέπονται περισσότερες από μία εγγραφές για κάθε `key`.
- Η λέξη κλειδί πρέπει να είναι ένας μη-μεταλλάξιμος τύπος (`str`, `int` κλπ). Όποτε, δεν μπορεί να είναι λίστα.

Παράδειγμα. [Δείτε και [σημειώσεις](#) Μ. Κολουντζάκη.] Δίνεται μία λίστα. Μετρήστε την συχνότητα κάθε στοιχείου που εμφανίζεται στη λίστα.

Ας υποθέσουμε μία λίστα L . Θα κατασκευάσουμε ένα λεξικό D στο οποίο οι λέξεις-κλειδιά θα είναι τα μοναδικά στοιχεία της L . Για κάθε λέξη-κλειδί θα βρούμε πόσες φορές αυτό εμφανίζεται στην L .

```
L = [1, 1, 2, 3, 2, 1, -1, 5, 5, 10, 1, 2, -1]

D = {} # Ξεκινάμε ένα λεξικό D (αρχικά κενό).
for key in L:
    if key in D.keys(): # Αν υπάρχει ήδη ως λέξη-κλειδί στο D τότε το
έχουμε ξαναδεί
        D[key] += 1      # και έχουμε μια παραπάνω εμφάνιση για το key
```

```
else:
    D[key] = 1      # Είναι η πρώτη φορά που το βλέπουμε.

print(D)          # Τυπώνουμε το λεξικό απλά

for key in D:     # Τυπώνουμε το λεξικό συστηματικά
    print(key, D[key])
```

Παρατήρηση: Μπορούμε να αναζητήσουμε εγγραφές στο λεξικό με την εντολή `if key in D.keys()` είτε και με την `if key in D`.

Συναρτήσεις και μέθοδοι για λεξικά

- `len(D)`: γίνει το αριθμό των εγγραφών.
- `str(D)`: δίνει όλο το λεξικό σε μία συμβολοσειρά.
- `del D`: σβήνει το `D`.
- `del D[key]`: σβήνει την εγγραφή με λέξη-κλειδί `key` από το `D`.
- `H = D.copy()`: αντιγράφει το `D` στο `H`.
- `D.update({key: object})`: προσθέτει εγγραφή στο `D`.
- `D.keys()`: οι λέξεις-κλειδιά του `D`.

Μελέτη

Βιβλιογραφία

1. **Σημειώσεις** Μ. Κολουντζάκη.
2. J.V. Guttag, *Υπολογισμοί και προγραμματισμός με την python* (Παράγραφος 5.1).
3. Δημήτριος Καρολίδης, *Μαθαίνετε εύκολα python* (Παράγραφος 4.3) (Εκδόσεις Καρολίδη, 2016).
4. Κ. Μαγκούτης, Χ. Νικολάου, *Εισαγωγή στον αντικειμενοστραφή προγραμματισμό με Python*, (Αποθετήριο "Κάλλιπος", 2016) - **Κεφάλαιο 6. Συμβολοσειρές, λίστες, πλειάδες, λεξικά.**

[Προηγούμενο](#)

[Επόμενο](#)