

Αντικειμενοστρεφής Προγραμματισμός - Python

Κ.Π. Γιαλούρης

Στόχοι του σημερινού μαθήματος

- Ανάλυση των μεταβλητών/κλάσεων τύπου
 - `tuple`
 - `dictionary`
 - `set`

tuple

Πλειάδα

Μία **tuple** (πλειάδα) είναι μία αλληλουχία όπως και η λίστα. Διαφέρει ως προς τη λίστα όσον αφορά στο γεγονός ότι το περιεχόμενό της δεν μπορεί να αλλάξει.

Δηλώνεται ως εξής:

```
a= ("Νικος", "Πέτρος", "Μαρία")
```

```
b= (4, 3, 5, 6, 2, -3)
```

```
c= (1, )
```

`tuple` Πλειάδα

Πλεονέκτημα έναντι της λίστας

Αυξάνει την ταχύτητα επεξεργασίας
Υπάρχει ασφάλεια λόγω μη δυνατότητα μεταβολής

tuple

Πλειάδα

Όπως και η λίστα η πλειάδα tuple υποστηρίζει τη διαχείριση των στοιχείων της μέσω δεικτών.

Διατηρεί τις ίδιες λειτουργίες με τη λίστα εκτός αυτών που αλλάζει το περιεχόμενό της.

tuple

Πλειάδα

Διαχείριση με δείκτες

Μέθοδοι όπως **index**

Συναρτήσεις **len**, **min**, **max**

Τελεστές

in

+

dictionary

Λεξικό

Μία μεταβλητή τύπου **dictionary** είναι συλλογή δεδομένων. Κάθε στοιχείο της συλλογής αυτής αποτελείται από δύο μέρη. Το πρώτο μέρος αναφέρεται ως κλειδί (**key**) και το δεύτερο μέρος ως τιμή (**value**).

dictionary

Λεξικό

Μία μεταβλητή τύπου `dictionary` είναι συλλογή δεδομένων. Κάθε στοιχείο της συλλογής αυτής αποτελείται από δύο μέρη. Το πρώτο μέρος αναφέρεται ως κλειδί (**key**) και το δεύτερο μέρος ως τιμή (**value**).

Το κλειδί χρησιμοποιείται για τον εντοπισμό μιας τιμής.

Σε κάθε κλειδί αντιστοιχεί μία τιμή.

dictionary

Μέθοδοι χειρισμού

clear διαγράφει το περιεχόμενο του λεξικού
`dictionary.clear()`

get επιστρέφει την τιμή ενός κλειδιού
`timi=dictionary.get(kleidi)`

items επιστρέφει όλα τα κλειδιά & τιμές ενός λεξικού
`dictionary.items()`

keys επιστρέφει όλα τα κλειδιά
`dictionary.keys()`

values επιστρέφει όλες τις τιμές ενός λεξικού
`dictionary.values()`

dictionary

Λεξικό -Δομή

Δημιουργία

```
ονομα_λεξικου= {}
```

```
ονομα_λεξικου= {κλειδι1:τιμή1, κλειδι1:τιμή1,  
κλειδι1:τιμή1,.....}
```

Ανάκτηση στοιχείου μέσω της:

```
μεταβλητή=ονομα_λεξικού[κλειδί]
```

```
μεταβλητή=ονομα_λεξικού.get(κλειδί)
```

Και το σύστημα επιστρέφει την τιμή εφόσον υπάρχει

dictionary

Λεξικό

```
phone_list={54182:"Γεωργοπουλος",54175:"Λιβάνιος",  
54183:"Κωτσιόπουλος"}
```

```
print(phone_list[54175])
```

dictionary

Λεξικό

Διαγραφή στοιχείου σε λεξικό

```
del Ονομα_λεξικού[κλειδί]
```

Έστω το λεξικό:

```
phone_list={54182:"Γεωργοπουλος", 54175:"Λιβάνιος",  
54183:"Κωτσιόπουλος"}
```

Θέλουμε να διαγράψουμε την εγγραφή με κλειδί 54183

```
del phone_list[54183]
```

dictionary

Λεξικό

Προσθήκη

Όνομα_λεξικού [κλειδί] = τιμή

Έστω το λεξικό:

```
phone_list={54182:"Γεωργοπουλος", 54175:"Λιβάνιος",  
54183:"Κωτσιόπουλος"}
```

Θέλουμε να προσθέσουμε μία εγγραφή που αφορά τον *Παππάς* με κλειδί το 54451

```
Phone_list[54451]= "Παππάς"
```

dictionary

Λεξικό

Έστω το λεξικό:

```
phone_list={54182:"Γεωργοπουλος", 54175:"Λιβάνιος",  
54183:"Κωτσιόπουλος"}
```

έλεγχος αν υπάρχει εγγραφή με κλειδί 54451

```
print(54451 in phone_list)
```

Παράδειγμα χειρισμού λεξικού

```
lexiko={ } # Δημιουργία κενού λεξικού {kleidi:value,...}
επαν=True
while επαν:
    kleidi=int(input("κλειδί "))
    if kleidi>0: # αν το κλειδί είναι θετικός αριθμός
        timi=float(input("timi "))
        if kleidi in lexiko: # αν υπάρχει εγγραφή στο λεξικό με
            # αυτό το κλειδί
            axia=lexiko.get(kleidi) # διαβάζεται η τιμή (value) της εγγραφής
            axia=axia+timi # αυξάνεται κατά το περιεχόμενο της
                # μεταβλητής timi
            lexiko[kleidi]=axia # η εγγραφή με το αντίστοιχο κλειδί
                # ενημερώνεται με τη νέα τιμή
        else:
            lexiko[kleidi]=timi # εισάγεται μία νέα εγγραφή
    else:
        επαν=False
for each in lexiko:
    print(each, lexiko.get(each)) # για κάθε κλειδί του λεξικού
        # εμφανίζει το κλειδί και την τιμή της εγγραφής
```

Set Σύνολο

Ένα σύνολο **set** είναι συλλογή δεδομένων από μοναδικές τιμές.

Τα στοιχεία ενός συνόλου **δεν είναι** διατεταγμένα

Τα στοιχεία ενός συνόλου **μπορεί να είναι** διαφορετικού τύπου

set Σύνολο

Δημιουργία

```
ονομα_συνόλου= set()
```

```
ονομα_συνόλου= set([στοιχείο-1,...στοιχείο-ν])
```

```
ονομα_συνόλου= set(string)
```

set Σύνολο

add	προσθέτει ένα στοιχείο σε σύνολο <code>myset.add(στοιχείο)</code>
update	προσθέτει σε υπάρχον σύνολο τα στοιχεία ενός string <code>myset.update(string)</code>
clear	διαγράφει το περιεχόμενο του συνόλου <code>myset.clear()</code>
discard	διαγράφει ένα συγκεκριμένο στοιχείο από το σύνολο <code>myset.discard(στοιχείο)</code>
remove	διαγράφει ένα συγκεκριμένο στοιχείο από το σύνολο <code>myset.remove(στοιχείο)</code>

set Σύνολο

union

Ένωση δύο συνόλων

```
newset=myset.union(oldset)
```

intersection

Τομή δύο συνόλων

&

```
newset=myset.intersection(oldset)
```

difference

Διαφορά δύο συνόλων

```
newset=myset.difference(oldset)
```

symmetric_difference

Συμμετρική διαφορά δύο συνόλων

```
newset=myset.symmetric_difference(oldset)
```

set Σύνολο

In έλεγχος ύπαρξης στοιχείου σε σύνολο

```
if x in myset
```

>= έλεγχος για υποσύνολο

issubset

<= έλεγχος για υπερσύνολο

issuperset

Παράδειγμα μεθόδων συνόλου ορισμός συνόλου

Έστω το πρόγραμμα:

```
a={1,2,3,4,5,6,7,8,9,10,11,12,13,14}  
print ('a=',a)  
b={4,5,6,7,8,9,10,11,12,15,16,17,18}  
print('b=',b)
```

Η εκτέλεση του θα έχει ως αποτέλεσμα :

```
a= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  
b= {4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18}
```

Παράδειγμα μεθόδων συνόλου ορισμός συνόλου από λίστα

Έστω το πρόγραμμα:

```
a=set([1,2,3,4,5,6,7,8,9,10,11,12,13,14,11,2])  
print ('a=',a)  
b=set([4,5,6,7,8,9,10,11,12,15,16,17,18])  
print('b=',b)
```

Η εκτέλεση του θα έχει ως αποτέλεσμα :

```
a= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  
b= {4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18}
```

Παράδειγμα μεθόδων συνόλου τομή συνόλων

Έστω το πρόγραμμα:

```
a={1,2,3,4,5,6,7,8,9,10,11,12,13,14}
b={4,5,6,7,8,9,10,11,12,15,16,17,18}
tomi_a_b=a.intersection(b)
print ('a τομή b=',tomi_a_b)
c={33,34,35,36}
print('c=',c)
tomi_a_c=a.intersection(c)
print ('a τομή c=',tomi_a_c)
```

Η εκτέλεση του θα έχει ως αποτέλεσμα :

```
Η εκτέλεση του θα έχει ως αποτέλεσμα :
a τομή b= {4, 5, 6, 7, 8, 9, 10, 11, 12}
c= {33, 34, 35, 36}
a τομή c= set()
```

Παράδειγμα μεθόδων συνόλου ένωση & διαφορά συνόλων

Έστω το πρόγραμμα:

```
a={1,2,3,4,5,6,7,8,9,10,11,12,13,14}
b={4,5,6,7,8,9,10,11,12,15,16,17,18}
enosi_a_b=a.union(b)
print ('a ένωση b=',enosi_a_b)
diafora_a_b=a.difference(b)
print('διαφορά a-b ',diafora_a_b)
sym_diafora_a_b=a.symmetric_difference(b)
print('συμμετρική διαφορά a-b ',sym_diafora_a_b)
```

Η εκτέλεση του θα έχει ως αποτέλεσμα :

```
a ένωση b= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
15, 16, 17, 18}
διαφορά a-b  {1, 2, 3, 13, 14}
συμμετρική διαφορά a-b  {1, 2, 3, 13, 14, 15, 16, 17, 18}
```


Παράδειγμα μεθόδων συνόλου υποσύνολα υπερσύνολα

Έστω το πρόγραμμα:

```
a={1,2,3,4,5,6,7,8,9,10,11,12,13,14}
print ('a=',a)
b={4,5,6,7,8,9,10,11,12,15,16,17,18}
print('b=',b)
d={1,2,3,4}
print("d=",d)
if d.issubset(a):
    print('το d υποσύνολο του a ')
if a.issuperset(d):
    print('το a υπερσύνολο του d ')
```

Η εκτέλεση του θα έχει ως αποτέλεσμα :

```
d= {1, 2, 3, 4}
το d υποσύνολο του a
το a υπερσύνολο του d
```

Παράδειγμα μεθόδων συνόλου

Έστω το πρόγραμμα:

```
a={1,2,3,4,5,6,7,8,9,10,11,12,13,14}
print ('a=',a)
b={4,5,6,7,8,9,10,11,12,15,16,17,18}
print('b=',b)
d={1,2,3,4}
print("d=",d)
if d.issubset(a):
    print('το d υποσύνολο του a ')
if a.issuperset(d):
    print('το a υπερσύνολο του d ')
arithmos=int(input ('δωσε ένα ακέραιο '))
if arithmos in a:
    print( "ο ",arithmos, " υπάρχει στο συνολο a")
else:
    print( "ο ",arithmos, " δεν υπάρχει στο συνολο a")
```

Παράδειγμα μεθόδων συνόλου

Η εκτέλεση του θα έχει ως αποτέλεσμα:

```
a= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}
```

```
b= {4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18}
```

```
a τομή b= {4, 5, 6, 7, 8, 9, 10, 11, 12}
```

```
c= {33, 34, 35, 36}
```

```
a τομή c= set()
```

```
a ένωση b= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,  
14, 15, 16, 17, 18}
```

```
διαφορά a-b {1, 2, 3, 13, 14}
```

```
συμμετρική διαφορά a-b {1, 2, 3, 13, 14, 15, 16, 17, 18}
```

```
d= {1, 2, 3, 4}
```

Παράδειγμα μεθόδων συνόλου έλεγχος ύπαρξης στοιχείου

Έστω το πρόγραμμα:

```
a={1,2,3,4,5,6,7,8,9,10,11,12,13,14}
b={4,5,6,7,8,9,10,11,12,15,16,17,18}
arithmos=int(input ('δώσε έναν ακέραιο='))
if arithmos in a:
    print( "ο ",arithmos, "υπάρχει στο σύνολο a")
else:
    print( "ο ",arithmos, "δεν υπάρχει στο σύνολο a")
```

Η εκτέλεση του θα έχει ως αποτέλεσμα :

```
δώσε έναν ακέραιο= 5
ο 5  υπάρχει στο σύνολο a
```

Άσκηση

Μια επιχείρηση αγροεφοδίων μέσω των αντιπροσώπων της διαθέτει στους αγρότες τρεις τύπους λιπασμάτων **A**, **B** και **Γ**. Να γραφεί πρόγραμμα το οποίο:

A. Για κάθε πώληση:

- Διαβάζει από την οθόνη:
 - (α) Το ονοματεπώνυμο του αντιπροσώπου,
 - (β) τον τύπο του λιπάσματος που πούλησε (**A**, **B** ή **Γ**) και
 - (γ) την αξία πώλησης του λιπάσματος.
- Υπολογίζει τη προμήθεια που δικαιούται ο αντιπρόσωπος από την πώληση του λιπάσματος. Η προμήθεια, για το λίπασμα **A** είναι 12% επί της αξίας πώλησης, για το λίπασμα **B** είναι 14% επί της αξίας πώλησης, ενώ για το λίπασμα **Γ** είναι 14.5 % επί της αξίας πώλησης.
- Στη συνέχεια το πρόγραμμα εμφανίζει στην οθόνη το ονοματεπώνυμο του αντιπροσώπου και την αντίστοιχη προμήθειά του.
- Η είσοδος των δεδομένων και ο υπολογισμός της προμήθειας επαναλαμβάνεται μέχρι να δοθεί ως ονοματεπώνυμο αντιπροσώπου η λέξη «**ΤΕΛΟΣ**».

B. Μετά το τέλος της εισαγωγής των δεδομένων, το πρόγραμμα:

- Για κάθε αντιπρόσωπο εμφανίζει στην οθόνη τη συνολική αξία προμήθειας.