ΑΣΚΗΣΕΙΣ – 10 Φεβρουαρίου 2021

Στο recording μπορείτε να βρείτε τη συζήτηση των ακόλουθων θεμάτων:

1. Δημιουργία dictionary με list comprehension:
   π.χ. mydict1 = {x*x: x for x in range(10)}
   Δημιουργούμε ένα dictionary όπου το key είναι ένας αριθμός (εδώ x^2, η τιμή είναι ο αριθμός x, και οι τιμές του x είναι στο διάστημα [0,10).
2. Δημιουργία dictionary με ανάμεικτα keys:
   π.χ. mydict2 = { 1:'banana', 'two':[1,2,5] }
3. Δημιουργία dictionary με μετατροπή tuple ή list σε dictionary:
   π.χ. mydict3(([1,2], [3,4])) ή mydict
   Η συνάρτηση dict() δέχεται 1 μόνο όρισμα οπότε για να εισαγάγουμε 2 ή περισσότερες lists θα πρέπει να τις περάσουμε ως list ή tuple.
4. Η δήλωση περισσότερο από μια φορές ενός key με τιμή οδηγεί στην επικράτηση της τιμής της τελευταίας δήλωσης:
   π.χ. mydict4 = {1:2, 1:3, 1:4, 2:4}
   Αν τυπώσουμε το mydict4 θα έχουμε: {1:4, 2:4}
5. Δήλωση ενός άδειου dictionary και εισαγωγή στοιχείων αργότερα:
   mydict5 = {}
   mydict5[1] ='dog' # το όνομα του key είναι 1 και η τιμή 'dog'
   mydict5[2] = 'cat'
6. Οι τιμές ενός dictionary λαμβάνονται δίνοντας το όνομα του dictionary και το επιθυμητό key μέσα σε [].
   Π.χ. από το 1ο παράδειγμα: mydict1[4] θα επιστρέψει 2. Προσοχή ότι αν δεν υπάρχει το key στο dictionary, θα επιστρέψει error.
7. Θα μπορούσαμε να πάρουμε την τιμή χρησιμοποιώντας τη μέθοδο get()
   Π.χ. mydict1.get(4) θα επιστρέψει 2. Αν δεν υπάρχει το key τότε δεν επιστρέφει τίποτα, αλλά αν βάλουμε την κλήση της συνάρτησης σε print(mydict1.get(9)) επιστρέφει none
8. Μπορούμε να αλλάξουμε την τιμή ενός key ή να επεκτείνουμε το dictionary με νέο key
   π.χ. mydict1[4] = 10 # αλλαγή της τιμής του key 4 από 2 σε 10 (αναφορά στο 1ο παράδειγμα)
   mydict1[10] = 1 #εισαγωγή του key 10 με τιμή 1
9. Μπορούμε να διαγράψουμε το dictionary ή στοιχεία του:
   π.χ. del mydict
   Αν το dictionary έχει στοιχεία mydict9={ 1:3, 2:4, 3:6, 4:8} μπορούμε να διαγράψουμε κάποιο item(ζεύγος) με την οδηγία: del mydict9[1]
10. Η συνάρτηση len(dictname) επιστρέφει το πλήθος των ζευγών key:value (τα items δηλαδή).

    π.χ len(mydict9) θα δώσει 4

11. Η συνάρτηση sorted μπορεί να χρησιμοποιηθεί για να ταξινομήσει ένα dictionary χωρίς να το τροποποιήσει. Η ταξινόμηση γίνεται σε αύξουσα σειρά με βάση τα keys του dictionary:
    π.χ. mydict11 = { 3:3, 1:1, 4:4}
    sorted(mydict11) επιστρέφει: [1, 3, 4]
    Είναι τύπου list όπως μπορούμε να δούμε δίνοντας την εντολή: type(sorted(mydict11))

12. Μπορούμε να εξετάσουμε αν υπάρχει κάποιο key στο dictionary:
    π.χ mydict12 = {1: 2, 2: 4, 3: 6}
    4 in mydict12 επιστρέφει False ενώ 2 in mydict12 επιστρέφει True

13. Μπορούμε να εφαρμόσουμε μια επαναληπτική διαδικασία για ένα dictionary:
    π.χ. mydict13 = {1:1, 3: 3, 4: 4}
    for i in mydict13:
    print(mydict13[i] *2)
    Θα τυπώσει: 2, 6, 8 το οποίο προκύπτει πολλαπλασιάζοντας την τιμή του κάθε key που στο παράδειγμα είναι [1],[3],[4] με 2.

14. Ένα dictionary key μπορεί να έχει ως τιμή ένα άλλο dictionary αλλά το key δεν μπορεί να είναι το ίδιο dictionary:
    π.χ. mydict14:{ 4: {1: 2, 2: 4}, 5: 10}
    mydict14[4] θα τυπώσει {1: 2, 2: 4}

15. Δημιουργία ενός νέου dictionary με τα keys ενός προϋπάρχοντος dictionary:
    π.χ. mydict15={3: 1, 4: 2, 1: 1, 5: 4, 6: 2}
    newdict = mydict15.fromkeys(mydict15.keys(), 0} ή διαφορετικά
    newerdict = mydict15.fromkeys({3, 4, 1, 5, 7}, 0)
    Χρήση της μεθόδου keys για να πάρουμε τα keys του mydict15 και ανάθεση της τιμής 0.

16. Εισαγωγή στοιχείου σε dictionary με την χρήση της μεθόδου get()
    π.χ. newdict[2] = newdict.get(2,0) + 1 # ψάχνει για το key 2, αν υπάρχει αυξάνει την τιμή του κατά 1 και τροποποιεί την υπάρχουσα τιμή, αν δεν βρεθεί τότε δημιουργείται το key και η αρχική τιμή του είναι 1 εφόσον η μέθοδος get επιστρέφει 0 ως τιμή.

17. Έστω ένα αριθμός ακέραιος N, να γράψετε ένα πρόγραμμα το οποίο δημιουργεί ένα dictionary που περιέχει (i, i*i) τέτοιο ώστε να είναι ένας ακέραιος στο διάστημα [1,N]. Υποθέστε ότι δίνετε N = 8, θα πρέπει να πάρετε {1: 1, 2: 4, 3: 9, 4:16, 5: 25, 6: 36, 7: 49, 8:64}.

```
N = int(inp())
d = dict()
for i in range(1,N+1):
    d[i] = i*i
print(d)
```

18. Να γράψουμε ένα πρόγραμμα το οποίο δέχεται μια σειρά από αριθμούς που χωρίζονται με , και δημιουργεί μια λίστα και ένα tuple με τα νούμερα. Δίνονται οι αριθμοί 34, 67, 55, 33, 12, 98.

```
Values = input()
L=values.split(“,”)
T = tuple(L)
```