

ΑΣΚΗΣΕΙΣ – 13 Φεβρουαρίου 2021

Στο recording μπορείτε να βρείτε τη συζήτηση των ακόλουθων θεμάτων:

0. Συζήτηση των ασκήσεων του 4^{ου} εργαστηρίου καθώς και των 2 πρώτων ασκήσεων του 3^{ου} quiz.

1. Ταξινόμηση dictionary με βάση τις τιμές και όχι τα keys

Έστω ένα dictionary `mydict={ "Anna":23, "Xrysa":18, "Antria":19, "Katerina":28, "Soti":26, "Nick":47, "Alex":38, "Marina":27, "Grigoris":45, "Kostas":62 }` το οποίο θέλουμε να ταξινομηθεί σύμφωνα με τις τιμές του.

Μπορούμε να ακολουθήσουμε τις ακόλουθες μεθόδους:

(a) Χρήση της συνάρτησης `sorted` και συνάρτησης `lambda`

`sortd = sorted(mydict.items(), key = lambda x: x[1])` που είναι ισοδύναμο με

`sortd = sorted(mydict.items(), key = lambda (k,v): v)`

(b) Χρήση της συνάρτησης `sorted` και της `list comprehension`

`sortd = sorted([(value,key) for (key,value) in mydict.items()])` και αν το θέλαμε σε φθίνουσα σειρά `sortd = sorted([(value,key) for (key,value) in mydict.items()], reverse=True)`

2. Αλλαγή χαρακτήρων σε γραμματοσειρά

(a) Για να αλλάξουμε κάποιο χαρακτήρα σε μια γραμματοσειρά χρησιμοποιούμε τη συνάρτηση `replace()`. Για παράδειγμα για να αλλάξουμε “,” σε space “ ”:

```
s = "this, is, a sentence"
s = s.replace(","," ")
print(s)
```

(b) Για να προσθέσουμε “,” σε κάποιον αριθμό χρησιμοποιούμε τη μέθοδο `str.format(number)` με τους χαρακτήρες “{:,}” ως `str`.

```
new_number = "{:,}".format(1000000)
print(new_number)
```

(c) Πως να αλλάξουμε σύμβολα σε μια string

```
Punctuations = `` ` ! ( ) 0 [ ] { } ; : " ' \ , < > . / ? @ # $ % ^ & * _ ~ ` ` ` ` # το \ είναι escape
line = "Hello!!!, he said -- and went."
no_punct = ""
for char in line:
    if char not in punctuations:
        no_punct = no_punct + char
print(no_punct)
```

(d) Αλλαγή string ζεύγους key-τιμής που διαχωρίζονται με κόμμα σε dictionary

```
intstring = "name = fotis, course = phyl45, branch=computational"
print("initial string", intstring)
res=dict(item.split("=") for item in intstring.split(","))
print("The final string is ", str(res))
```

Διαφορετικά με `map` και `lambda`

```
res = dict(map(lambda x: x.split('='), intstring.split(',')))
print("The final string is ", str(res))
```

(e) Θα μπορούσαμε να γράψουμε ένα πολυώνυμο σε μορφή dict όπου τα keys είναι οι δυνάμεις και οι σταθεροί όροι οι τιμές:
poly={0: -1, 2: 1, 7: 3.5}

```
def eval_poly_dict(poly, x):  
    sum = 0.0  
    for power in poly:  
        sum += poly[power]*x**power  
    return sum
```

Ως lists:

```
P = [-1, 0, 1, 0, 0, 0, 0, 3]  
def eval_poly_dict(poly,x):  
    sum = 0  
    for power in range(len(poly)):  
        sum +=poly[power] * x**power  
    return sum
```