

ΗΜΥ01Κ06
Επιστημονικός Προγραμματισμός με Python



6+1 ερωτήσεις επανάληψης για τη Διάλεξη 06
*Δομημένοι Τύποι Δεδομένων (συνέχεια): Σύνολα, Λεξικά
Διαχείριση Σφαλμάτων*

Φθινόπωρο 2025

Ερώτηση 1/7

Πως συμβολίζεται ένα κενό σύνολο και πως ένα κενό λεξικό;

Κενό σύνολο : `set()`

Κενό λεξικό : `{}`

ή ενναλακτικά `dict()`

Ερώτηση 2/7

Οι μέθοδοι `.remove()` και `.discard()` χρησιμοποιούνται και οι δυο για την αφαίρεση ενός συγκεκριμένου στοιχείου από ένα σύνολο
Ποια η διαφορά τους;

```
fruits = { 'apple', 'banana', 'cherry' }  
fruits.remove('banana')  
print(fruits) ⇒ {'cherry', 'apple'}
```

```
fruits = { 'apple', 'banana', 'cherry' }  
fruits.discard('banana')  
print(fruits) ⇒ {'apple', 'cherry'}
```

Αν το στοιχείο που θέλουμε να αφαιρέσουμε δεν υπάρχει, τότε το `remove` **επιστρέφει μήνυμα λάθους**

ενεργοποιεί μια εξαίρεση

Αν το στοιχείο που θέλουμε να αφαιρέσουμε δεν υπάρχει, τότε το `discard` **ΔΕΝ** επιστρέφει μήνυμα λάθους

Ερώτηση 3/7

Έστω `f = {'orange', 'banana', 'cherry'}` το αρχικό σύνολο για όλα επόμενα παραδείγματα. Τι εκτυπώνεται σε καθένα από αυτά;

```
f.add('apple')
print(f)
{'banana', 'orange', 'apple', 'cherry'}
```

```
f.update('apple')
print(f)
{'e', 'orange', 'banana', 'cherry', 'a', 'l', 'p'} γιατί;;
```

```
f.update(('apple'))
print(f)
{'e', 'banana', 'cherry', 'orange', 'a', 'l', 'p'} και πάλι γιατί;;
```

```
f.update(('apple',))
print(f)
{'cherry', 'apple', 'banana', 'orange'}
```

Ερώτηση 4/7

Τι είναι τα παγωμένα -ή παγωμένα- σύνολα (frozen sets);

Σύνολα που αφού δημιουργηθούν, *δεν μπορούν πια να τροποποιηθούν* προσθέτοντας ή αφαιρώντας άλλα στοιχεία (*είναι αμετάβλητα αντικείμενα*)

Τα παγωμένα σύνολα δημιουργούνται με την συνάρτηση `frozenset()`

```
x = frozenset( { 1, 2, 3} ) #Μετατροπή κανονικού συνόλου σε frozenset
print(x) ⇒ frozenset ( { 3, 1, 2} )
```

```
x = frozenset( [ 1, 2, 3] ) # Μετατροπή λίστας σε frozenset
print(x) ⇒ frozenset ( { 1, 3, 2} )
```

```
x = frozenset( range(0:5) ) #Δημιουργία frozenset από range
print(x) ⇒ frozenset ( {4, 1, 3, 0, 2} )
```

```
x = frozenset( 'Hello' ) # Δημιουργία frozenset από συμβολοσειρά
print(x) ⇒ frozenset ( { 'e', 'H', 'o', 'l' } )
```

Ερώτηση 5/7

Λεξικά στην Python. Σωστό ή λάθος;

- ✓ Οι *τιμές* ενός λεξικού μπορεί να είναι οποιοδήποτε αντικείμενο της Python
- ✓ Τα *κλειδιά* ενός λεξικού αποτελούνται μόνο από *αμετάβλητα* στοιχεία
- ✗ Ένα λεξικό *μπορεί να περιέχει* το ίδιο κλειδί περισσότερες από μια φορές
- ✓ Ένα *παγιωμένο σύνολο* μπορεί να αποτελέσει κλειδί ενός λεξικού
- ✗ Στο `for x in my_dict:` το `x` διατρέχει όλες τις *τιμές* του `my_dict`
- ✗ Η εντολή `del mycar['model']` εκχωρεί στο κλειδί `'model'` την τιμή `None`

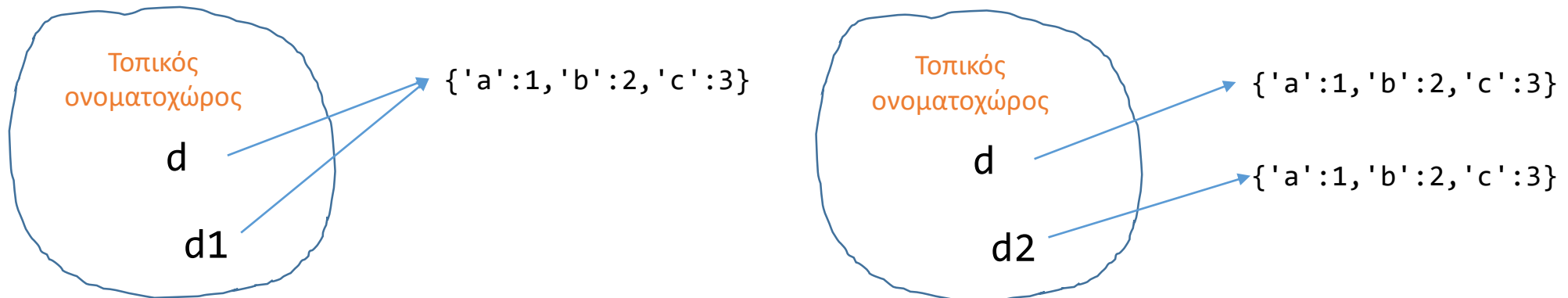
Ερώτηση 6/7

Αν `d` είναι λεξικό, σε τι διαφέρουν οι εντολές

`d1 = d` και `d2 = dict(d)`

`d1 = d` : δημιουργεί *ένα ακόμα όνομα* για το λεξικό `d`

`d2 = dict(d)` : δημιουργεί *ένα αντίγραφο* του λεξικού `d`



Ερώτηση 6/7

Ποια είναι η απλούστερη δυνατή μορφή της δομής `try/except`

`try:`

μπλοκ_εντολών

το τμήμα εντολών του προγράμματος στο οποίο επιδρά το `try / except`

`except:`

εντολές_χειρισμού_εξαιρέσεων

εκτελούνται αν εγερθεί **οποιαδήποτε** εξαίρεση

Εναλλακτική μορφή

`try:`

μπλοκ_εντολών

`except Exception as e:`

`print(e)` # Σύντομη περιγραφή του σφάλματος (εξαίρεσης)

υπόλοιπες_εντολές_χειρισμού_εξαιρέσεων (προαιρετικά)

Bonus Ερώτηση 7/7

Γνωρίζοντας ότι όλα τα κλειδιά ενός λεξικού *είναι μοναδικά*,
θα μπορούσε να είναι σωστή μια έκφραση της μορφής

`x = d['a']['a']` αν το `d` είναι λεξικό;

Ναι, θα μπορούσε, ως εξής:

Η τιμή που αντιστοιχεί στο κλειδί `'a'` του `d` είναι ένα άλλο
λεξικό που (*τυχαία*) έχει και αυτό ένα κλειδί `'a'`

```
>>> d1={'a':1, 'b':2, 'c':3}
>>> d2={'a':100, 'b':200, 'c':300}
>>> d={'a':d1, 'b':d2}
>>> d['a']
{'a': 1, 'b': 2, 'c': 3}
>>> d['a']['a']
1
>>>
```