



Σύνολα (Sets)



Χαρακτηριστικά Συνόλων

- ▶ Είναι παρόμοια με την μαθηματική έννοια του συνόλου.
- ▶ Είναι μεταβαλλόμενος τύπος δεδομένων.
- ▶ Περιέχει αμετάβλητου τύπου μοναδικά στοιχεία.
 - ▶ Άρα στοιχεία που είναι λίστες δεν επιτρέπονται.
- ▶ Δεν υπάρχουν διπλότυπα στοιχεία.
- ▶ Δεν υπάρχει διάταξη.
- ▶ Η χρήση τους βοηθά πολύ στην ταχύτητα των αναζητήσεων.

Δημιουργία συνόλων

```
IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> set1={5,10,15,20}
>>> set1
{10, 20, 5, 15}
>>> set2={1,2,3,4,5,5,4,3,2,1}
>>> set2
{1, 2, 3, 4, 5}
>>> set3={'Lamia',3,5,(6,7,8)}
>>> set3
{'Lamia', 3, 5, (6, 7, 8)}
>>> set4={'Lamia',3,5,[6,7,8]}
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    set4={'Lamia',3,5,[6,7,8]}
TypeError: unhashable type: 'list'
>>> |
```

Πράξεις με σύνολα

- ▶ Ισχύουν οι τελεστές 'in' και 'not in'
- ▶ Ισχύουν οι γνωστοί τελεστές σύγκρισης
- ▶ Τελεστής ένωσης συνόλων: '|' (ισχύει και η μέθοδος `union()`)
- ▶ Τελεστής τομής συνόλων: '&' (ισχύει και η μέθοδος `intersection()`)
- ▶ Τελεστής διαφοράς δύο συνόλων: '-' (ισχύει και η μέθοδος `difference()`)
- ▶ Τελεστής συμμετρικής διαφοράς (μη κοινά στοιχεία): '^' (ισχύει και η μέθοδος `symmetric_difference()`)
- ▶ Για τις συγκρίσεις μπορούν να χρησιμοποιηθούν και οι μέθοδοι `issuperset()` και `issubset()`
- ▶ Το σύνολο που προκύπτει από τις πράξεις αυτές, ακόμη κι αν έχει το ίδιο όνομα, είναι διαφορετικό αντικείμενο.

Πράξεις με σύνολα – Επαυξημένοι τελεστές

Τελεστής	Μέθοδος
$ =$	<code>update()</code>
$\&=$	<code>intersection_update()</code>
$-=$	<code>difference_update()</code>
$\wedge=$	<code>symmetric_difference_update()</code>

Προσθήκη και διαγραφή στοιχείων σε σύνολα

```
>>> set1={10,11,12,13,14,15}
>>> set1.add(16)
>>> set1
{16, 10, 11, 12, 13, 14, 15}
>>> set1.add([1,2,3])
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    set1.add([1,2,3])
TypeError: unhashable type: 'list'
>>> set1.remove(16)
>>> set1
{10, 11, 12, 13, 14, 15}
>>> set1.discard(15)
>>> set1
{10, 11, 12, 13, 14}
>>> set1.pop()
10
>>> set1.clear()
>>> set1
set()
>>>
```

- ▶ Δεν επιτρέπεται να προστεθεί λίστα γιατί είναι μεταβαλλόμενη.
- ▶ Η μέθοδος `discard()` δεν επιστρέφει σφάλμα αν δεν υπάρχει το στοιχείο που προσπαθούμε να διαγράψουμε.
- ▶ Η μέθοδος `pop()` διαγράφει ένα τυχαίο στοιχείο από το σύνολο.

Άλλες λειτουργίες με σύνολα

```
>>> set1={1,2,3}
>>> set2=set1.copy()
>>> set2
{1, 2, 3}
>>> set3=set1
>>> id(set1)
2236274446848
>>> id(set2)
2236274446624
>>> id(set3)
2236274446848
>>> list1=[4,5,6]
>>> set1.isdisjoint(list1)
True
>>> len(set1)
3
>>>
```

➤ Αντιγραφή.

➤ Έλεγχος αν υπάρχει τομή.

➤ Πλήθος στοιχείων συνόλου.

Σύνολα και επαναλήψεις

```
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>> set1={1,2,3,4,5}  
>>> for i in set1:  
    print(i)  
    |
```

Ln: 78 Col: 1



Παγωμένα Σύνολα (Frozen Sets)

- ▶ Στα απλά σύνολα δεν μπορεί ένα σύνολο να είναι μέλος ενός άλλου συνόλου.
- ▶ Τα παγωμένα σύνολα έχουν όλες τις ιδιότητες των απλών συνόλων, εκτός της μεταβλητικότητάς τους.
 - ▶ Έτσι μπορούμε να δημιουργούμε σύνολα από λίστες.
- ▶ Δημιουργούνται με τη χρήση της μεθόδου `frozenset()`.
- ▶ Δεν μπορούμε να χρησιμοποιήσουμε μεθόδους των απλών συνόλων, όπως `remove()`, `add()`, κλπ.
- ▶ Μπορούμε να χρησιμοποιήσουμε τους τελεστές ένωσης, τομής, τους επαυξημένους τελεστές, κλπ.

Παγωμένα Σύνολα

```
IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> f1=frozenset({1,2,4,5,3,7,9,2,3,5})
>>> f1
frozenset({1, 2, 3, 4, 5, 7, 9})
>>> list1=list(range(0,50,5))
>>> list1
[0, 5, 10, 15, 20, 25, 30, 35, 40, 45]
>>> set1=frozenset(list1)
>>> set1
frozenset({0, 35, 5, 40, 10, 45, 15, 20, 25, 30})
... --+1501
```

Παγωμένα Σύνολα

```
>>> set2=frozenset("This is a string")
>>> set2
frozenset({'a', 'r', 'n', 'T', 't', 's', 'h', ' ', 'q', 'i'})
```

```
>>> set3=frozenset({i for i in list1[3:8]})
>>> set3
frozenset({35, 20, 25, 30, 15})
>>>
```

Σημείωση: τα στοιχεία των συνόλων δεν έχουν διάταξη. Η αποθήκευσή τους γίνεται από το διερμηνευτή με σειρά δική του, που διαφέρει από αυτή με την οποία έγινε η εισαγωγή τους στο σύνολο.